Encontrando falhas em aplicações web baseadas em flash

Wagner Elias
Gerente de Pesquisa e Desenvolvimento
Conviso IT Security

Tópicos

- Um pouco sobre flash
- Vulnerabilidades
- Como fazer direito
- Conclusões

Um pouco sobre Flash

Quem desenvolve em flash?

- A maioria das aplicações desenvolvidas em flash são:
  - Peças publicitárias desenvolvidas por agências de propaganda
  - Interferências visuais criadas por designers

O que eles esquecem?

- Flash é *Client-Side* e pode ser decompilado apresentando o fonte em ActionScript

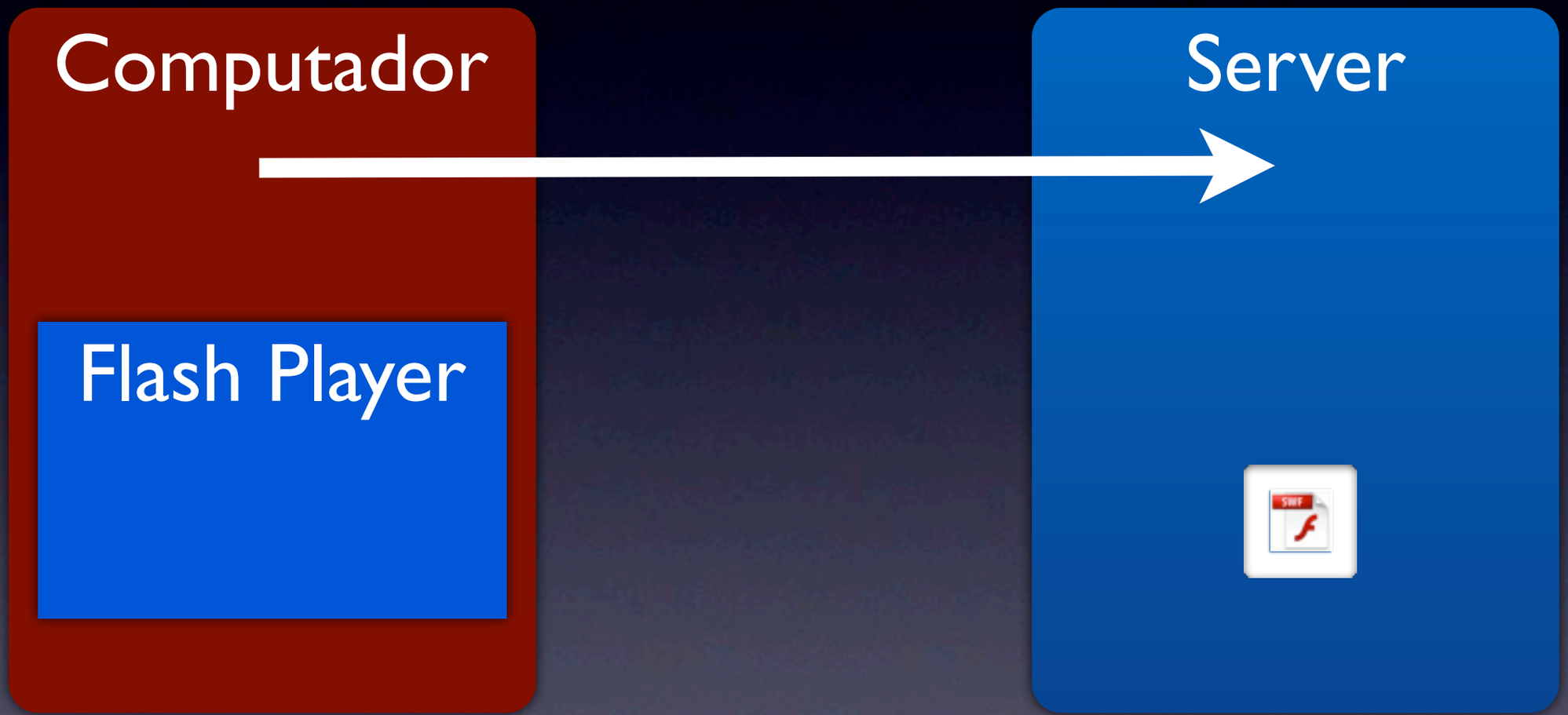
Flash in Client-Side

Computador

Flash Player

Server

Flash in Client-Side



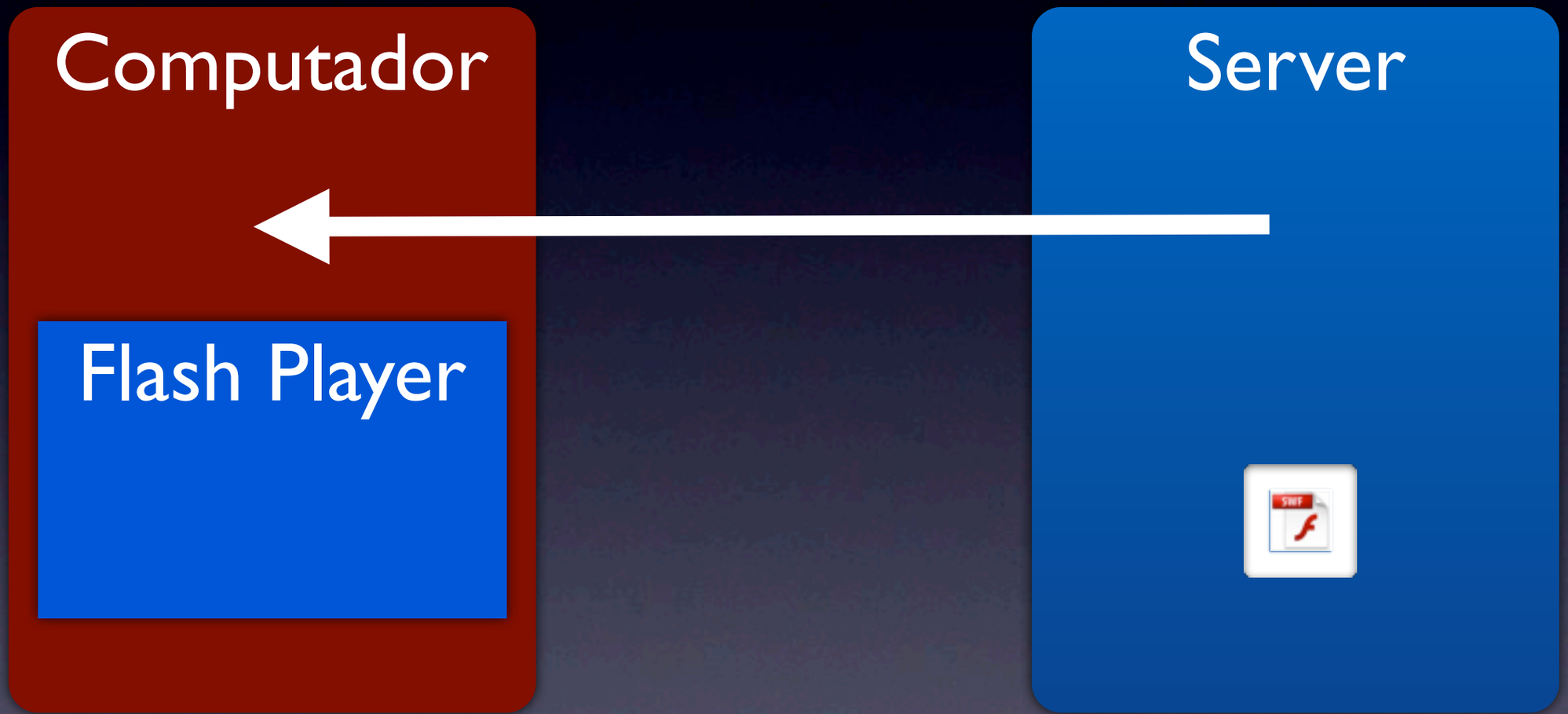
Flash in Client-Side

Computador

Flash Player

Server

Flash in Client-Side



Flash in Client-Side

Computador

Flash Player

Server

Flash in Client-Side

Computador

Flash Player

Server

Flash in Client-Side

Computador

Flash Player

Server

filetype:swf

- Milhares de telas de login com usuário e senha em *Client-Side*
  - filetype:swf inurl:login OR inurl:secure OR inurl:admin

- Milhares de arquivos crossdomain.xml sem controles de segurança
  - filetype:xml inurl:crossdomain

Vulnerabilidades

Information Leakage

- Como o swf roda na máquina do usuário, qualquer informação, segredo no código está disponível para qualquer um que decompilar o arquivo swf

Flash Decompile

AS Compila e gera SWF

CS3 / MTASC

SWF decompila gera AS

SWFScan / Flare

DEMO

Cross Site Scripting (XSS)

- Assim como qualquer aplicação o flash também precisa ter seus inputs tratados

```
class VulnerableMovie {
    static var app : VulnerableMovie;
    function VulnerableMovie() {
        _root.createTextField("tf",0,100,100,640,480);

        if (_root.userinput1 != null) {
            getURL(_root.userinput1);
        }

        _root.tf.html = true; // default is safely false
        _root.tf.htmlText = "Hello " + _root.userinput2;

        if (_root.userinput3 != null ) {
            _root.loadMovie(_root.userinput3);
        }
    }
    static function main(mc) {
        app = new VulnerableMovie();
    }
```

DEMO

Cross Site Scripting (XSS)

- Assim como qualquer aplicação o flash também precisa ter seus inputs tratados

```
class VulnerableMovie {
    static var app : VulnerableMovie;
    function VulnerableMovie() {
        _root.createTextField("tf",0,100,100,640,480);

        if (_root.userinput1 != null) {
            getURL(_root.userinput1);
        }

        _root.tf.html = true; // default is safely false
        _root.tf.htmlText = "Hello " + _root.userinput2;

        if (_root.userinput3 != null ) {
            _root.loadMovie(_root.userinput3);
        }
    }
    static function main(mc) {
        app = new VulnerableMovie();
    }
```

DEMO

Cross Site Scripting (XSS)

- Assim como qualquer aplicação o flash também precisa ter seus inputs tratados

```
class VulnerableMovie {
    static var app : VulnerableMovie;
    function VulnerableMovie() {
        _root.createTextField("tf",0,100,100,640,480);

        if (_root.userinput1 != null) {
            getURL(_root.userinput1);
        }

        _root.tf.html = true; // default is safely false
        _root.tf.htmlText = "Hello " + _root.userinput2;

        if (_root.userinput3 != null ) {
            _root.loadMovie(_root.userinput3);
        }
    }
    static function main(mc) {
        app = new VulnerableMovie();
    }
```

DEMO

Cross Site Scripting (XSS)

- Assim como qualquer aplicação o flash também precisa ter seus inputs tratados

```
class VulnerableMovie {
    static var app : VulnerableMovie;
    function VulnerableMovie() {
        _root.createTextField("tf",0,100,100,640,480);

        if (_root.userinput1 != null) {
            getURL(_root.userinput1);
        }

        _root.tf.html = true; // default is safely false
        _root.tf.htmlText = "Hello " + _root.userinput2;

        if (_root.userinput3 != null ) {
            _root.loadMovie(_root.userinput3);
        }
    }
    static function main(mc) {
        app = new VulnerableMovie();
    }
```

DEMO

Cross Site Scripting (XSS)

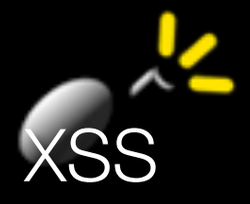
- Assim como qualquer aplicação o flash também precisa ter seus inputs tratados

```
class VulnerableMovie {
    static var app : VulnerableMovie;
    function VulnerableMovie() {
        _root.createTextField("tf",0,100,100,640,480);

        if (_root.userinput1 != null) {
            getURL(_root.userinput1);
        }

        _root.tf.html = true; // default is safely false
        _root.tf.htmlText = "Hello " + _root.userinput2;

        if (_root.userinput3 != null ) {
            _root.loadMovie(_root.userinput3);
        }
    }
    static function main(mc) {
        app = new VulnerableMovie();
    }
```



DEMO

Cross Site Scripting (XSS)

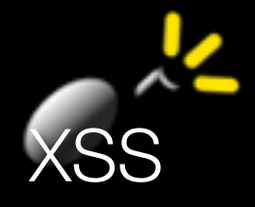
- Assim como qualquer aplicação o flash também precisa ter seus inputs tratados

```
class VulnerableMovie {
    static var app : VulnerableMovie;
    function VulnerableMovie() {
        _root.createTextField("tf",0,100,100,640,480);

        if (_root.userinput1 != null) {
            getURL(_root.userinput1);
        }

        _root.tf.html = true; // default is safely false
        _root.tf.htmlText = "Hello " + _root.userinput2;

        if (_root.userinput3 != null ) {
            _root.loadMovie(_root.userinput3);
        }
    }
    static function main(mc) {
        app = new VulnerableMovie();
    }
```



DEMO

Cross Site Scripting (XSS)

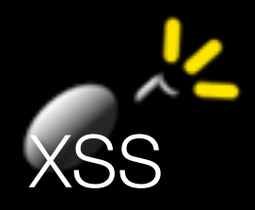
- Assim como qualquer aplicação o flash também precisa ter seus inputs tratados

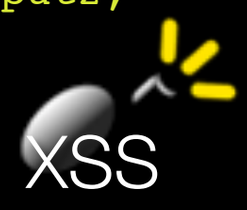
```
class VulnerableMovie {
    static var app : VulnerableMovie;
    function VulnerableMovie() {
        _root.createTextField("tf",0,100,100,640,480);

        if (_root.userinput1 != null) {
            getURL(_root.userinput1);
        }

        _root.tf.html = true; // default is safely false
        _root.tf.htmlText = "Hello " + _root.userinput2;

        if (_root.userinput3 != null ) {
            _root.loadMovie(_root.userinput3);
        }
    }
    static function main(mc) {
        app = new VulnerableMovie();
    }
```

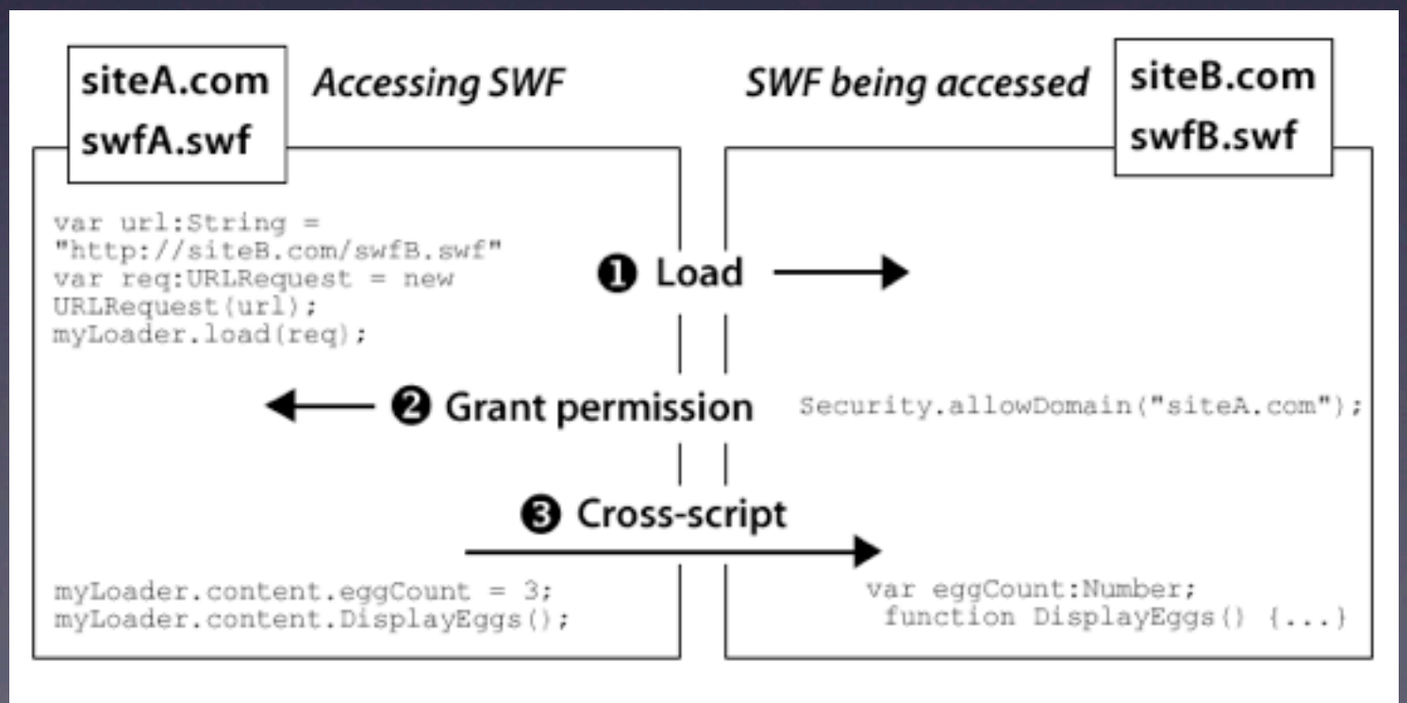




DEMO

Crossdomain

- Por padrão o Flash Player (a partir da versão 7) implementa um SandBox
  - Apenas com um crossdomain.xml é possível comunicar com outras aplicações



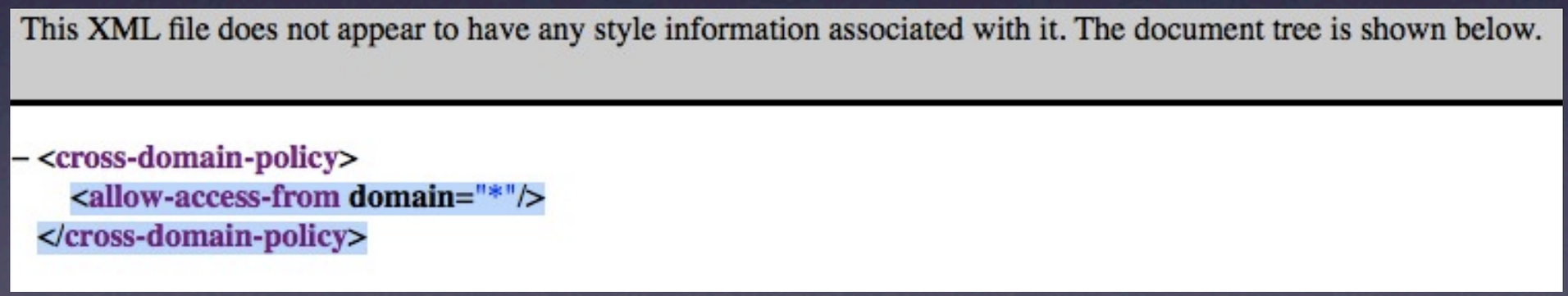
Crossdomain Abuse

- O problema é inserir o crossdomain.xml usando o wildcard padrão e sem determinar quais endereços podem se comunicar

DEMO

Crossdomain Abuse

- O problema é inserir o crossdomain.xml usando o wildcard padrão e sem determinar quais endereços podem se comunicar



DEMO

Crossdomain Abuse

- O problema é inserir o crossdomain.xml usando o wildcard padrão e sem determinar quais endereços podem se comunicar

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <cross-domain-policy>
    <allow-access-from domain="*"/>
  </cross-domain-policy>
```

Permite que qualquer um acesse o conteúdo

DEMO

Como fazer direito

Flash Security Facts

- Não insira informações sensiveis em flash
- Valide todos os inputs
- Defina quais domínios podem acessar o conteúdo no crossdomain.xml
- Se for inevitável o flash, criptografar o ActionScript (SWF Encrypt)

Conclusões

Conclusões

- A Adobe adora a Apple
- ActionScript é uma linguagem de programação e deve ser tratada como tal
- É preciso definir e implementar controles de segurança como em qualquer aplicação web

Web 2.0?

Perguntas?