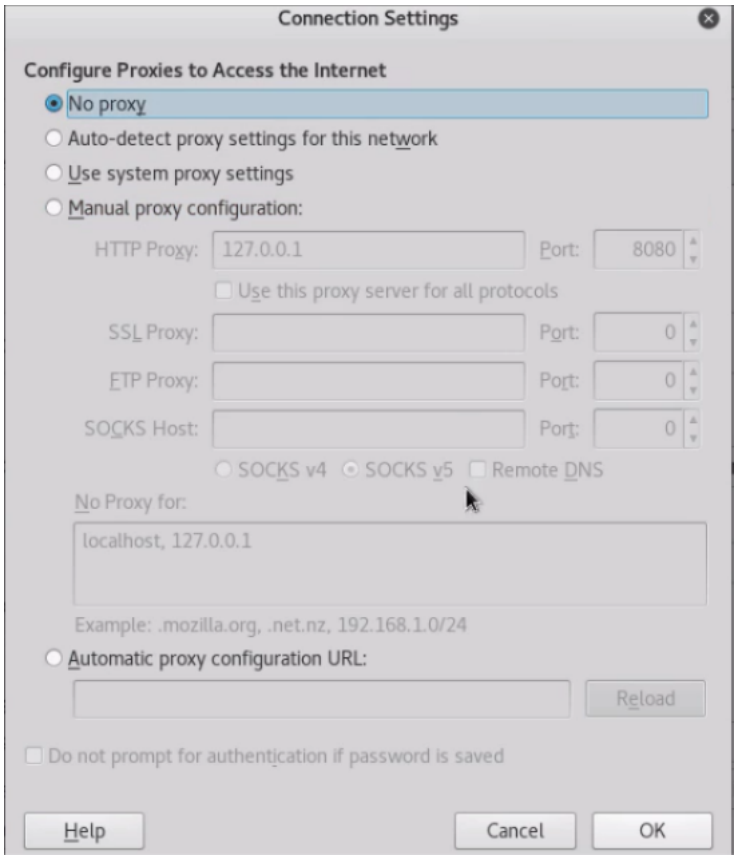
01

Cross site scripting

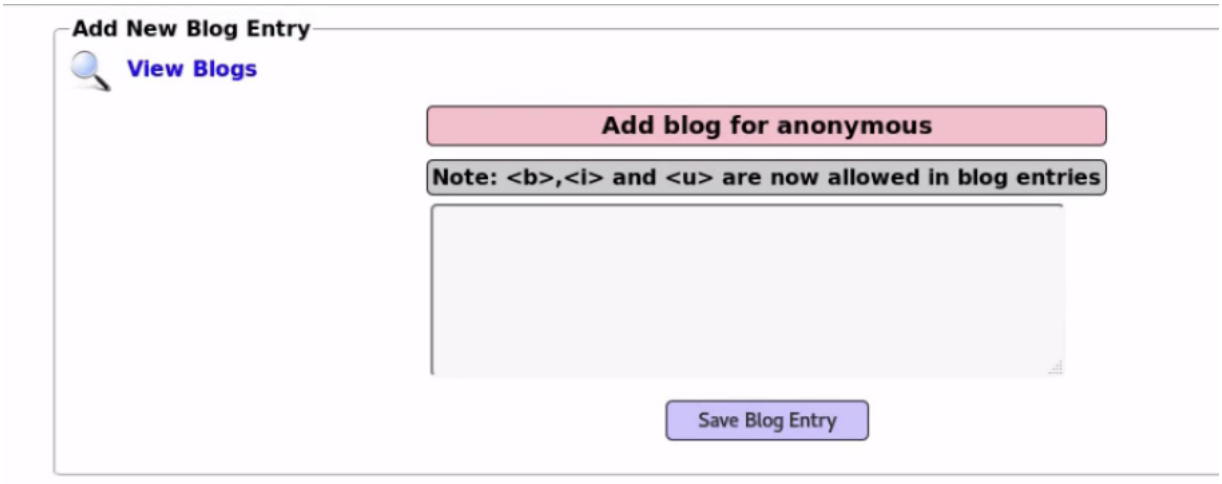
Transcrição

Já verificamos como funciona um ataque por meio de listas e do **Burb Suite**, agora, vamos entender outros mecanismos de invasão utilizados. Como o **Burb Suite** não será mais utilizado retornaremos as configurações originais do browser. Para isso, clicaremos no ícone de três linhas no Menu do navegador e "Preferences > Advanced > Networking > Settings" e na tela deixaremos marcado "No proxy" e damos um "Ok":



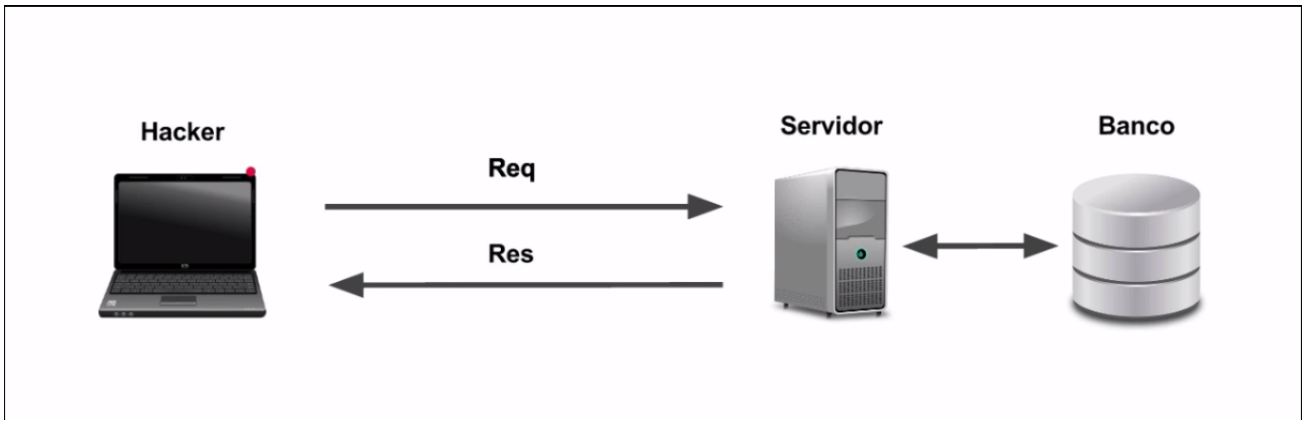
Com as configurações originais podemos prosseguir!

O site da *Multillidae* possui uma seção de Blog. O próximo passo é inserir uma mensagem dizendo que gostamos muito do site. Portanto, seguimos pelo caminho "OWASP 2O13 > A3 - Cross Site Scripting(XSS) > Persistent(Second Order) > Add to your blog". O trecho do blog tem a seguinte aparência:

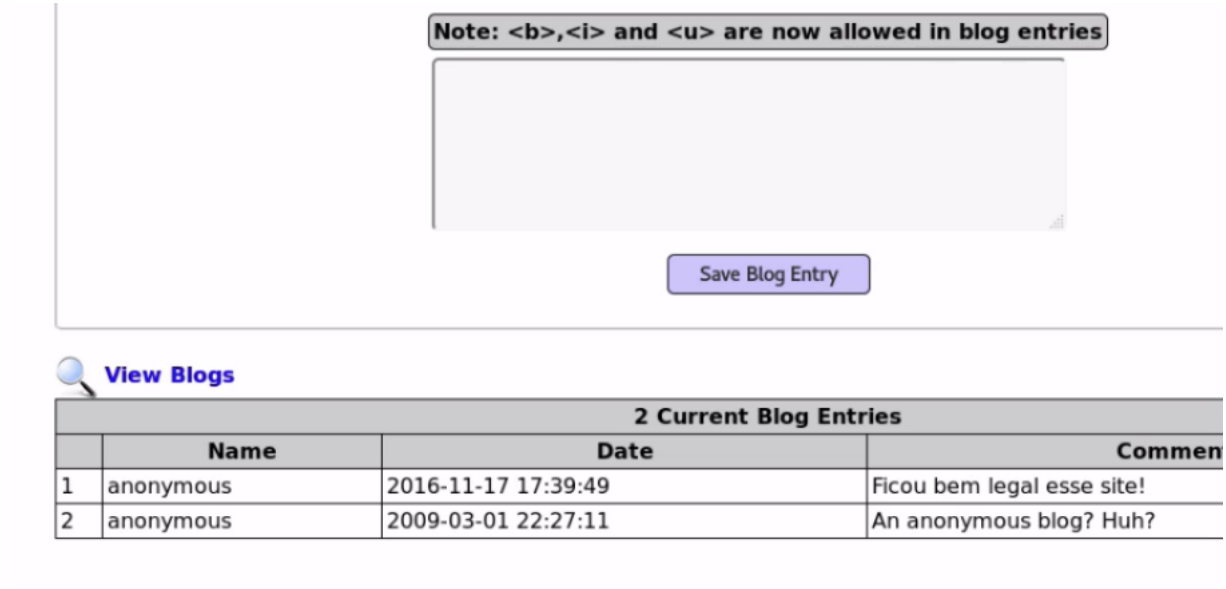


Antes de qualquer coisa, vamos entender o que acontece ao inserir uma mensagem no blog. Quando o hacker apertar o botão "Save blog Entry" é enviada uma requisição para o servidor. Na requisição são passadas as informações contidas na mensagem, por exemplo , "O site está muito legal".

O servidor guarda as informações no Banco de Dados e o banco faz uma consulta em todas as mensagens já enviadas e devolve para o hacker as mensagens atualizadas, inclusive, com a que ele próprio inseriu.



Então, retornando a página da *Multillidae* podemos escrever a mensagem, clicar no botão *Save Blog Entry* e é esperado que ela apareça na tela:

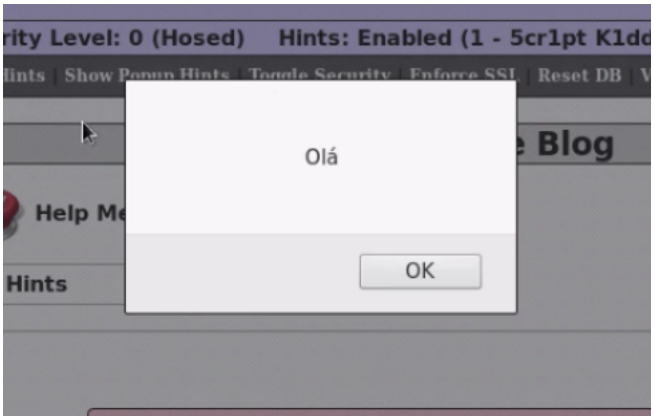


Falamos anteriormente sobre Inserção de código SQL e nesta aula utilizaremos a mesma lógica: Será que o desenvolvedor filtrou os caracteres da mensagem ou será que conseguimos inserir um código?

Vamos testar! Nesta aula, tentaremos inserir trechos de código *javascript* no espaço destinado aos comentários, portanto, nós colocaremos um `script` e um código de caixa de *alert* com a palavra `Olá` :

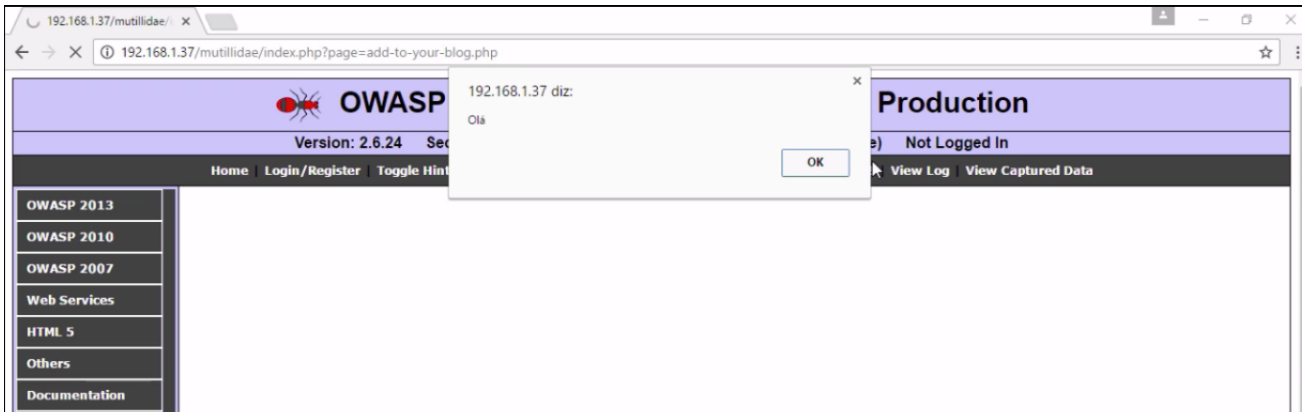
```
<script>
alert("Olá");
</script>
```

Assim, caso o *script* seja aceito como uma mensagem ele também será guardado no banco de dados e uma vez que a mensagem é salva, espera-se que seja devolvida pelo servidor. Para testar basta inserir o código na caixa de comentário, clicar em "Save blog Entry" e temos o seguinte:



Ou seja, o `script` do *box alert* foi retornado ao usuário e o código foi aceito como mensagem válida.

Vamos sair da máquina do hacker e acessar o site da *Multillidae* como usuário normal. No papel de usuário comum, nós acessamos o Blog e ao checar na página nos deparamos com o seguinte:



Justamente o *pop up* que o *hacker* escreveu! Ou seja, o que o hacker adicionou aparece em outro computador também! Mas, por qual motivo isso ocorre?

Como mencionado, o `script` que o hacker inseriu foi interpretado como uma mensagem válida pelo servidor e foi persistido no Banco de Dados, assim, quando uma vítima ou outro usuário acessar a página, o servidor faz a consulta das mensagens e traz o `script` interpretado e manda para a vítima, o usuário.

Um *pop up* escrito "Olá" parece inofensivo, entretanto, usando o código *javascript,* nós podemos inserir qualquer informação e esse tipo de ataque no qual inserimos um código visando manipular outros usuários é conhecido como **Cross Site Scripting***(XSS) e pelo fato de persistir no Banco de Dados, ele é classificado também como **Persistent**.