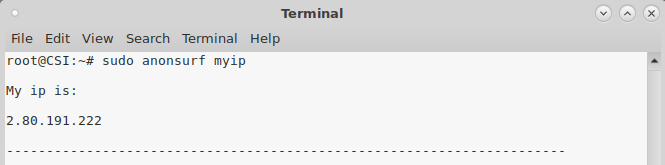
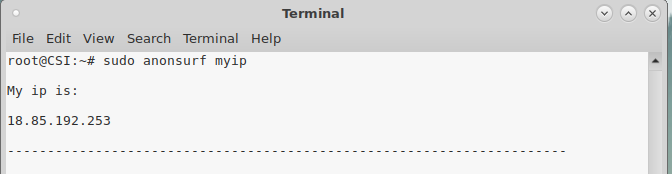
## Experiência 1.1

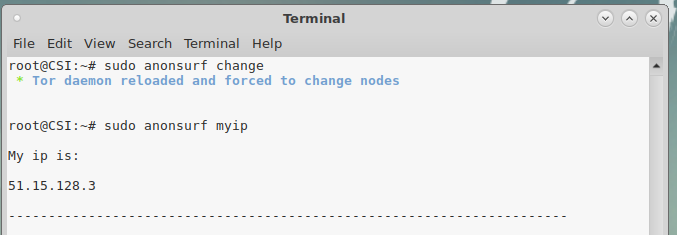
Inicialmente podemos verificar que o nosso IP verdadeiro é:



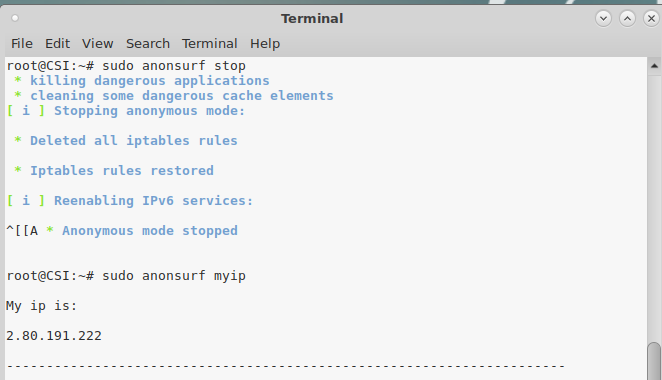
Após inicializarmos o *anonsurf*  e executarmos novamente o mesmo comando reparamos que já nos encontramos com um ip diferente.



Se efetuarmos o comando *sudo anonsurf change* é percetível que o nosso ip é novamente alterado.



Por último se pararmos o *anonsurf* reparamos que o ip volta para o mesmo valor inicial.



## Pergunta 1.1

### Pergunta 1

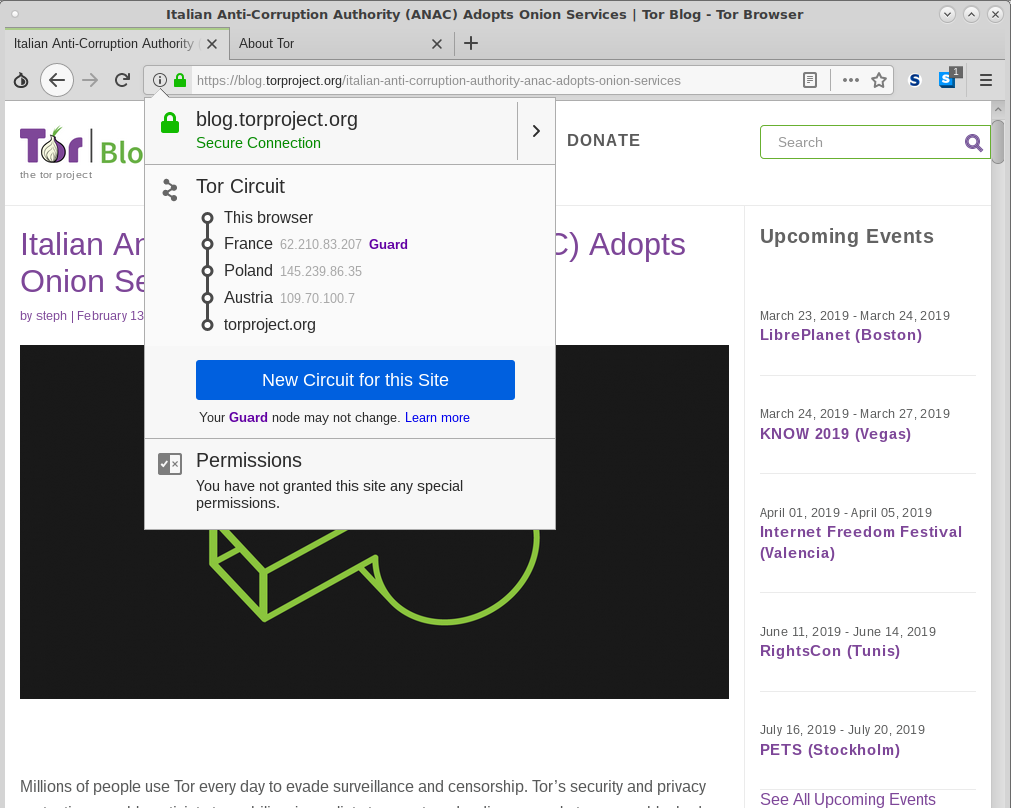
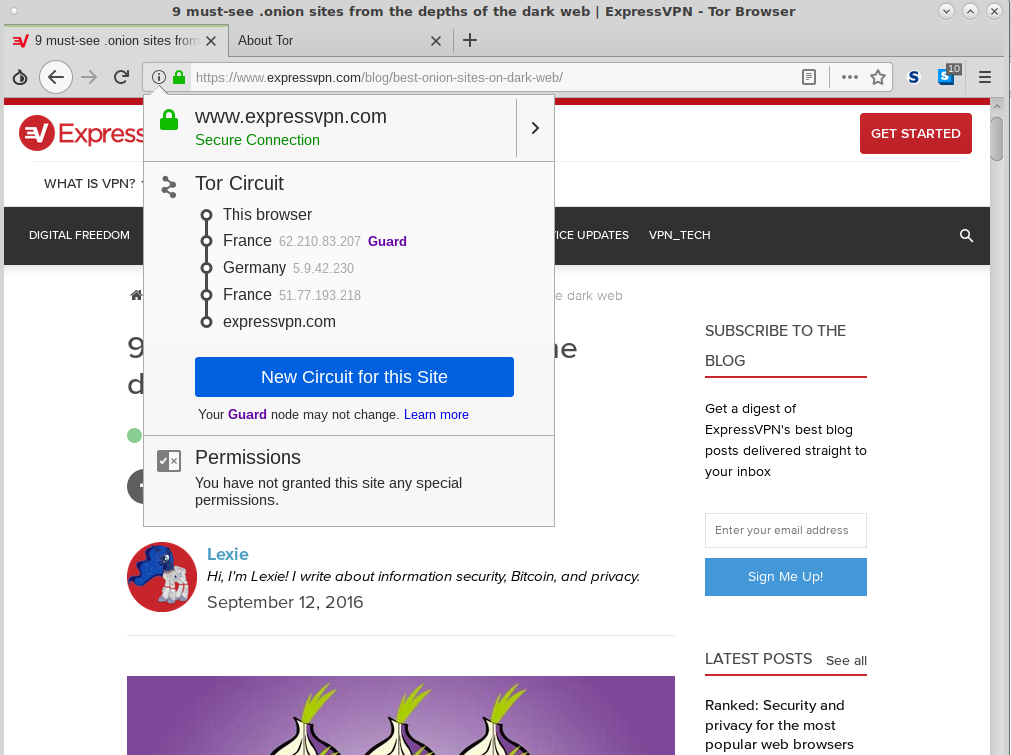
Não, ao executar esse comando apenas temos garantia que seremos conectados a uma máquina aleatória, podendo ou não estar situada nos EUA.

### Pergunta 2

O protocolo TOR cria automaticamente circuitos, constituídos por 3 *Onion Routers*, e altera este circuito de minuto a minuto. Assim sendo, concluímos que devido ao facto de não controlarmos os circuitos que são criados pelo protocolo, aliado ao facto de este circuito estar em constante transformação é impossível garantir a localização geográfica dos nós constituintes, logo não podemos garantir que estamos localizados no EUA.

## Experiência 1.2

O objetivo desta experiência era mostrar que para cada nova conexão feita através do Tor Browser é criado um novo *path* para o redireccionamento da informação trocada entre o browser e o website visitado. Existe uma constante que se trata do *Entry Point* na camada de redireccionamento. Como é possível ver nos prints foram feitos 3 saltos antes do primeiro pacote chegar ao servidor, escondendo assim a origem do pedido. Também podemos ver que só foram realizados 3 saltos porque o servidor não está a usar o protocolo Tor (url normal, sem acabar em .onion e a não existência de relays).

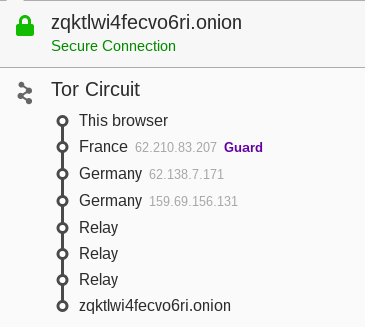
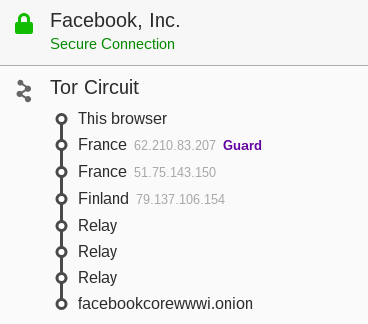


## Pergunta 1.2

### Pergunta 1

Seguem-se abaixo os circuitos gerados para os dois sites. À esquerda está representado o circuito gerado para o site <http://zqktlwi4fecvo6ri.onion/wiki/index.php/Main_Page> e à direita está representado o circuito gerado para o site <https://www.facebookcorewwwi.onion>.

Podemos observar que ambos os circuitos apresentam os três típicos *Onion Routers* escolhidos pelo *Onion* Proxy do cliente e adicionalmente três *Onion Routers* escolhidos pelo fornecedor do serviço.

### Pergunta 2

Anteriormente verificamos que para um utilizador Alice aceder de forma anónima a um serviço disponibilizado pelo utilizador Bob, necessita de estabelecer um circuito, geralmente de três saltos, e de três *Onion Routers* (ORs), até ao serviço. Este procedimento garante que ninguém consegue identificar o serviço ao qual a Alice está a aceder dado que o OR1 só conhece o OP da Alice e o OR2, o OR2 só conhece o OR1 e o OR3 e por fim o OR3 só conhece o OR2 e o OP do serviço (Bob), ou seja, o OR1 consegue apenas identificar que a Alice está a utilizar a rede TOR mas não consegue retirar mais informações, o OR2 não consegue retirar qualquer informação pois apenas reencaminha informação cifrada do OR1 para o OR3 e o OR3 apesar de possuir a informação decifrada, não consegue identificar qual é a origem da comunicação.

O mecanismo descrito acima por si só apenas anonimiza os utilizadores dos serviços, não fornecendo qualquer anonimato aos fornecedores dos mesmos. Deste modo, o protocolo TOR apresenta também uma funcionalidade que permite anonimizar os serviços, utilizando para isso um circuito de normalmente seis saltos, sendo os três primeiros saltos referentes aos três *Onion Routers* do *Directory Server* escolhidos pelo *Onion Proxy* do utilizador que pretende aceder ao serviço (parte do circuito escolhida pelo utilizador do serviço), enquanto que os restantes três saltos correspondem aos *Onion Routers* escolhidos pelo *Onion Proxy* do fornecedor do serviço (parte do circuito escolhida pelo fornecedor do serviço).

Para um utilizador fornecer um serviço anónimo XYZ.onion, começa por criar um par de chaves, sendo a chave pública responsável por identificar o serviço. De seguida escolhe alguns pontos de introduçãoe indica-os ao *Directory Server*, criando um circuito TOR com cada um destes pontos de introdução.

Por sua vez, para um utilizador utilizar um serviço anónimo, pode consultar os seus detalhes, nomeadamente chave pública e pontos de introdução, através do *Directory Server*. Posteriormente seleciona um *Onion Router* para ser um ponto de *rendezvous point* (RP) e constrói um circuito TOR até esse RP, fornecendo-lhe um *rendezvous cookie*. Após este processo de preparação do RP, o utilizador abre um *stream* anónimo até um dos pontos de introdução do serviço e envia-lhe informação sobre o RP, a *rendezvois cookie* e a sua parte do segredo para a troca de chaves Diffie-Hellman, ficando este encarregue de reencaminhar a mensagem até ao serviço XYZ.onion (através do circuito criado anteriormente).

Se o fornecedor do serviço XYZ.onion pretender comunicar com utilizador, cria um circuito TOR até ao RP, enviando-lhe a *rendezvous cookie*, a segunda parte do segredo para a troca de chaves Diffie-Hellman e uma *hash* da chave de sessão partilhada.

No final de todo este processo, o RP conecta o circuito do utilizador do serviço com o circuito do fornecedor do serviço, originando assim um circuito com, normalmente, seis ORs: três escolhidos pelo utilizador, sendo o terceiro o RP, e três escolhidos pelo fornecedor. Deste modo, tanto o utilizador como o fornecedor conhecem apenas a sua metade do circuito e o RP não consegue identificar nem o utilizador, nem o fornecedor, nem os dados que reencaminha, uma vez que estes estão cifrados.

Nas imagens apresentadas na alínea anterior, três dos nodos são *relay nodes* pois os respetivos ORs foram escolhidos pelo *Onion Proxy* do fornecedor do serviço. Deste modo, o utilizador do serviço apenas conhece o circuito até ao RP (parte do circuito escolhida por ele), não tendo qualquer informação sobre os restantes ORs do mesmo.