UNIVERSIDADE DE ÉVORA

Engenharia Informática
Segurança informática

Tor
The onion router

Professor: Pedro Patinho

João Cavaco nº42470
João Abel nº 42941

2021

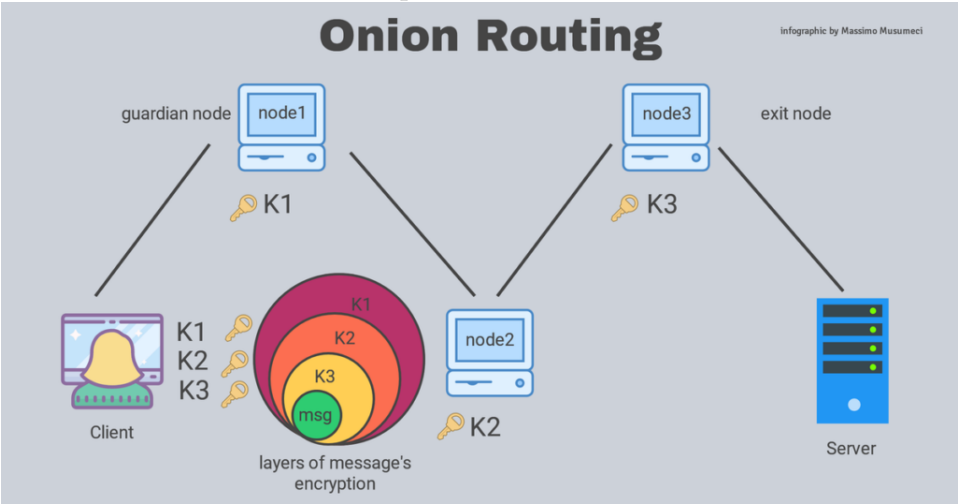
Índice

1 The Onion Router

Tor é um software *open-source* que permite a navegação anónima na web através da encriptação e encaminhamento do tráfego TCP através da **Tor network** que é uma rede overlay que contém milhares de **Tor relays** (servidores voluntários de retransmissão também referidos como nós) que são escolhidos aleatoriamente durante o encaminhamento do tráfico o que dificulta imenso a observação de qualquer tipo de padrão por parte de um observador externo que tenha como objectivo determinar informações sobre um ou vários utilizadores.

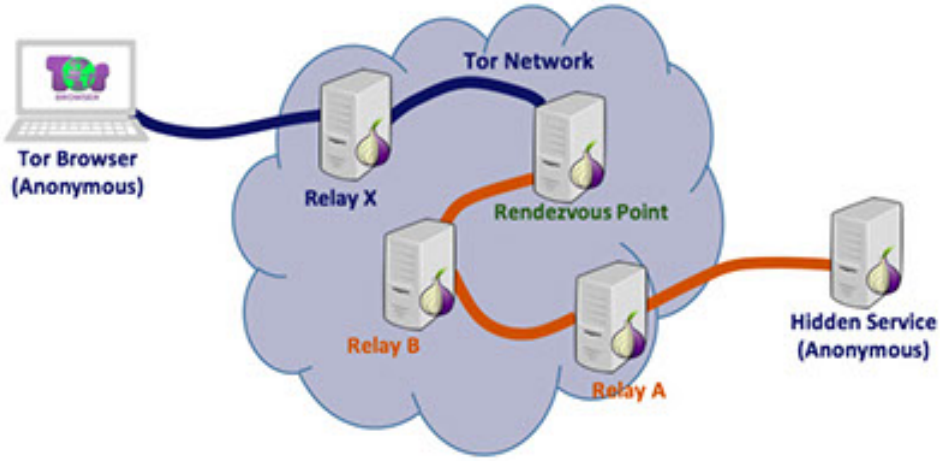
1.1 Onion routing

Tor utiliza o protocolo *onion routing*, que é uma técnica para comunicação anónima e segura através de uma rede pública. No *onion routing* as mensagens são encapsuladas em várias camadas de encriptação (normalmente 3 camadas) utilizando um sistema de chaves criptográficas simétricas (AES). Após a encriptação, é estabelecida uma conexão aleatória a um dos nós de entrada publicamente listados e a mensagem é enviada para o nó de entrada. Após a recepção da mensagem, é desencriptada uma das camadas que permite ao nó atual descobrir qual o próximo nó para o qual enviar a mensagem que ainda contém 2 camadas de encriptação, este passo vai ser repetido 2 vezes para as camadas ainda restantes mas após a desencriptação da mensagem final o ultimo nó (nó de saída) envia a mensagem para o servidor destinatário. O servidor recebe a mensagem processa a informação e envia uma resposta encriptada de volta pelo *circuito* criado previamente estabelecido ou através de um novo *circuito* constituído por novos nós.



2 Onion services

Os servidores configurados para receber ligações de entrada apenas através do Tor são chamados *onion services* ou *hidden services*. Em vez de revelar o endereço IP de um servidor um onion service é acedido através do seu *onion adress*, geralmente através do Tor Browser. A rede Tor compreende estes endereços procurando as suas chaves públicas correspondentes e pontos de introdução a partir de uma hash table que se encontra dístribuida dentro da rede Tor. Como todo o tráfego para os *onion services* é encaminhado através da rede Tor, a ligação a um *onion service* é encriptada em ambas as pontas e não está sujeita a escutas.



3 Segurança e anonimidade

Embora o Tor encripte dados entre o computador do utilizador e os servidores na rede Tor e dentro da rede Tor, não encripta a parte final da ligação entre o nó de saída e o servidor de destino. Como resultado, é possível que alguém analise o tráfego entre o nó de saída e os servidores de destino. Uma vez que a lista completa dos nós de saída do Tor está disponível ao público, qualquer tráfego não encriptado que saia dos nós de saída é provável que seja monitorizado de perto. Para combater este fator basta realizar apenas pedidos HTTPS de modo a encriptar todas as informações enviadas entre o nó de saída e o servidor destino.

Outra preocupação de segurança é quando o nó de entrada e saída existem ambos no mesmo sistema autónomo da Internet (AS), ou seja, o mesmo operador de rede é dono ambos os endereços IP. Se for esse o caso, é possível que esse operador de rede utilize técnicas de análise estatística e temporal para determinar a origem do tráfego. Esta técnica é difícil de executar, por isso normalmente só é possível que os governos consigam. Além disso, pode ser dispendioso, pelo que não é geralmente uma preocupação, exceto para alvos de alto valor.

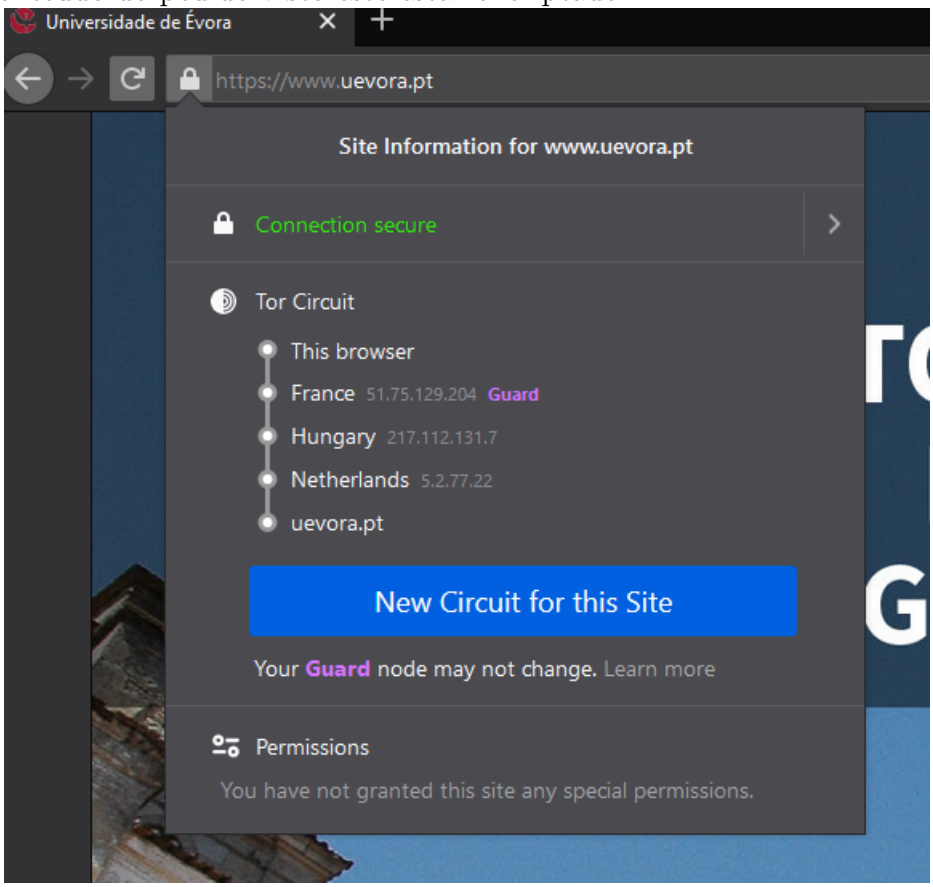
Outros métodos também podem ser usados para os quebrar o anonimato. Se por acaso estiver a navegar num website comprometido utilizando o Tor com o JavaScript ativado, é possível que o intruso determine quem é baseado nos movimentos do seu rato. A maioria das pessoas move o rato de uma forma distinta que pode ser usada para correlacionar uma sessão de navegação tor com uma sessão de navegação normal.

4 Utilização prática

Para aceder à rede Tor é recomendado a utilização do Tor browser (uma versão do Mozilla Firefox bastante modificada).

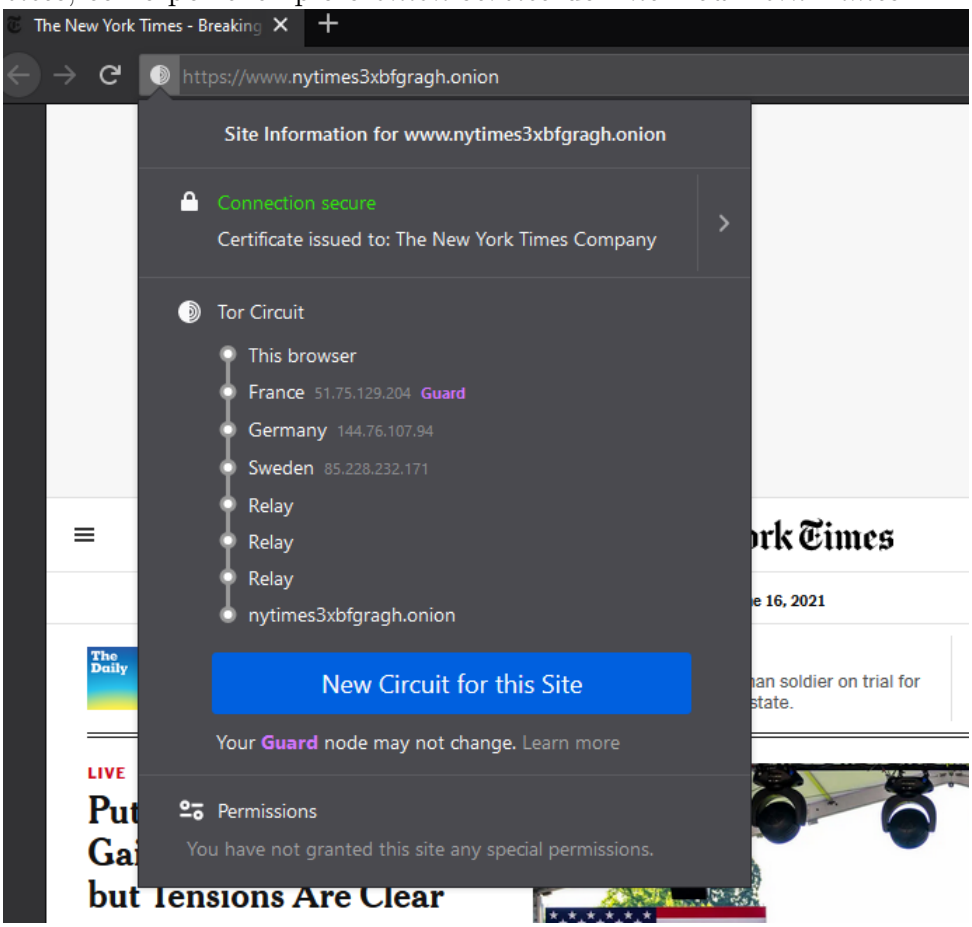
4.1 Aceder a serviços normais

Podemos aceder ao site da Universidade de Évora através da rede Tor tal como fazemos num browser normal porém como é possível observar na imagem seguinte o nosso tráfego foi encaminhado por 3 nós diferentes que se encontram em países distintos de modo a ocultar quem está a aceder ao site e o pedido realizado foi um pedido *HTTPS* o que significa que mesmo que alguém estivesse a escutar todo o tráfego nesse nó não iria conseguir observar o conteúdo do pedido visto este estar encriptado.



4.2 Onion services

É possível aceder a *onion services* utilizando o Tor browser, apenas é necessário saber o seu endereço. Felizmente existem bastante recursos online que nos permite encontrar facilmente uma enorme quantidade de *onion services*, como por exemplo o *onion service* do *The New York Times*.



4.3 Tor no desenvolvimento de software

Ao encaminhar o tráfego para a rede Tor podemos substituir pedidos *HTPP*/HTTPS que se encontrem, por exemplo, numa API que estejamos a desenvolver para tornar os pedidos anónimos ou para realizar pedidos a *onion services*.

4.4 Exemplo prático

No trabalho prático da disciplina de Redes de Computador no ano letivo anterior o objetivo era desenvolver uma aplicação que permitisse consultar a qualidade do ar baseado em sensores presentes nas várias cidades do país. Usando uma a API openAQ podemos desenvolver um sensor que a cada X minutos realiza um pedido e armazena o valor numa base de dados. Se tivermos bastantes sensores a realizarem imenso pedidos corremos o risco do nosso IP ser bloqueado e para impedir que isso aconteça vamos encaminhar todos os pedidos pela rede Tor.

4.4.1 Requisitos

- Ambiente GNU\Linux
- Tor (em ambientes Debian **sudo apt install tor**)
- Python3 e os respetivos módulos
  - requests
  - fake_useragents
  - stem
- editar o ficheiro torrc de acordo com a documentação (descomentar as 3 linhas na figura seguinte) para que se passe a utilizar uma porta controlo

```
## The port on which Tor will listen for local connections from Tor
## controller applications, as documented in control-spec.txt.
ControlPort 9051
## If you enable the controlport, be sure to enable one of these
## authentication methods, to prevent attackers from accessing it.
HashedControlPassword [redacted]
CookieAuthentication 1
```

4.4.2 Execução

- Inicializar o tor através do terminal
- executar o código Python que está disponível na secção seguinte.

Ao executarmos estes passos podemos observar o nosso endereço IP real e o endereço do servidor proxy pelo qual o nosso pedido vai ser encaminhado.

5 Código

```
# Imports necessarios
import requests
from fake_useragent import UserAgent
from stem import Signal
from stem.control import Controller
from pprint import pprint
import json
import os

# Mostrar o nosso ip verdadeiro
real_ip = requests.get("http://httpbin.org/ip").text
print(f"This is your ip address: {real_ip}")

# Encaminhar o nosso pedido atraves de proxies
session = requests.session()
session.proxies = {
    "http": "socks5://127.0.0.1:9050",
    "https": "socks5://127.0.0.1:9050",
}
# Randomizar o user-agent do nosso pedido
session.headers = {"User-Agent": UserAgent().random}

# Obter uma nova identidade Tor
# Ver documentacao oficial para uma explicacao em
   detalhe
with Controller.from_port(port=9051) as controller:
    controller.authenticate()
    controller.signal(Signal.NEWNYM)

# Mostrar o ip do servidor proxy
fake_ip = session.get("http://httpbin.org/ip").text

if real_ip == fake_ip:
    # impedir que pedidos sejam efetuados com o
       nosso ip verdadeiro
    print("Real IP detected")
    print("Please check if Tor is running and if a
       proxy is being used")
```

```
    exit(1)
else:
    # realizar o pedido encaminhando o trafego para
      um proxy
    print(f"This is the ip address used to make the
      request: {fake_ip}")
    api_url =
      "https://api.openaq.org/v2/countries/PT"
    response_text =
      json.loads(session.get(api_url).text)
    pprint(response_text)
```

6 Bibliografia

- Onion Routing
- Computerphile - How does Tor work?
- massmux - How does tor really work
- Tor the second generation onion router
- AES
- Computerphile - AES
- Avoid getting blocked when web scraping