

# Entendendo a *Deep Web*

Simeí Tabordes Gonçalves

<sup>1</sup>Fatec Senac  
Pelotas – RS – Brazil  
tabordes@gmail.com

**Resumo.** *Este artigo tem por objetivo descrever o que é a Deep Web. Para ser compreendida corretamente este artigo vai separar e mostrar como funcionam as tecnologias que formam a Deep Web como um todo, que corresponde a um conjunto de elementos onde o conteúdo é na maioria das vezes anônimo mas as tecnologias para acessar esse conteúdo são livres para qualquer usuário utilizar.*

**Abstract.** *This article aims to describe what is Deep Web To be properly understood apart and this article will show how the technologies that form the Deep Web as a whole work, which corresponds to a set of elements which content is most sometimes anonymous but the technology to access that content are free to use any user.*

## 1. Introdução

*Deep Web* é o termo foi o termo criado para descrever os conteúdos que só pode ser acessados através da rede TOR(The Onion Network), que é uma rede de computadores onde os roteadores não dão informações detalhadas das redes percorridas para ir de uma ponta a outra durante uma conexão entre hosts e servidores.

Os roteadores são na verdade hosts configurados para serem pontos de encontro, entrada ou saída das conexões oriundas de hosts e servidores. O objetivo final desse intrincado processo é manter o anonimato dos hosts, servidores e usuários. Ninguém na rede TOR consegue rastrear uma conexão da forma convencional como se faz na Web comum. O projeto inicialmente foi patrocinado pelo laboratório de pesquisa da marinha americana, mas hoje é patrocinado pelo governo americano, governo sueco e outras organizações não governamentais.

O orçamento anual de 2012 para o projeto da rede TOR foi de 2 milhões de dólares. Por causa do anonimato que a rede TOR proporciona para os usuários, a *Deep Web* acaba sendo muito procurada por pessoas com interesses diversos nem sempre voltados para o bem comum, mas também é procurada por pessoas que moram em países sob forte censura ou que simplesmente desejam esconder um conteúdo ou disponibilizá-lo de forma secreta.

O foco deste artigo é a tecnologia utilizada para o acesso a *Deep Web* e não entrar no mérito do seu conteúdo, até por que a Web comum também disponibiliza conteúdos de teor discutível que não são encontrados em indexadores de busca comuns. Infelizmente a *Deep Web* ficou famosa através do conteúdo criminoso e não pela sua utilidade e sofisticação tecnológica.

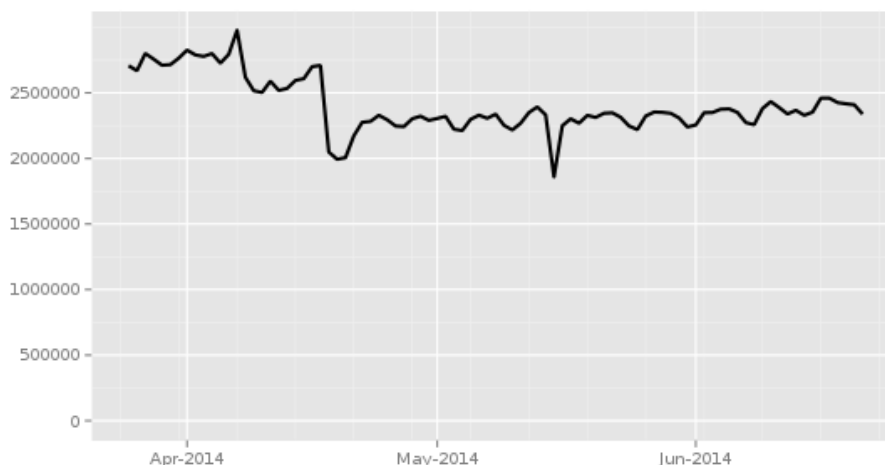
## 2. A rede TOR

### 2.1. Origens

A origem do nome TOR vem do acrônimo "*The Onion Router*". Onion em inglês significa cebola. A cebola foi escolhida por que possui diversas camadas, que simbolizam os diversos roteadores que são utilizados para gerar as camadas de isolamento que separam os hosts e servidores na rede TOR.

### 2.2. Objetivo

O objetivo maior é tornar anônimo o endereço ip do host ou servidor dentro da rede através da encriptação dos pacotes que são repassados entre os roteadores de forma anônima através de chaves criptografadas. Dessa forma o host só conhece o ip do primeiro roteador por onde o pacote passa. Daí pra frente os pacotes passam por um circuito criado pela rede TOR que escolhe aleatoriamente os roteadores para fechar o circuito.



**Figura 1.** (Número de usuários conectados segundo o site <https://metrics.torproject.org/users.html>)

### 2.3. TOR Browser

É o *browser* oficial da rede TOR. É uma versão modificada do conhecido Firefox da Mozilla. Roda em mídias removíveis e possui versões para os sistemas operacionais Linux, Windows e Mac OS X. O TOR Browser assim que é aberto inicia os processos para rotear tráfego para a rede TOR. Quando é terminada a sessão, são deletados dados de privacidade como *cookies* e histórico. As portas de saída utilizadas pelo TOR Browser são 25, 119, 135-139, 445, 563, 1214, 4661-4666, 6346-6429, 6699, 6881, 6999.

### 2.4. Hidden Service Protocol

O HSP é o protocolo responsável por criar os circuitos dentro da rede TOR. Esses circuitos são criados para ser estabelecida uma conexão entre *host* e servidor de forma anônima. O protocolo encripta chaves que são trocadas entre roteadores que fecham um segmento do circuito. Um *Hidden Service* precisa antes de tudo se anunciar na rede TOR. Ele escolhe aleatoriamente três roteadores, cria rotas até eles, e pede para serem pontos de entrada.

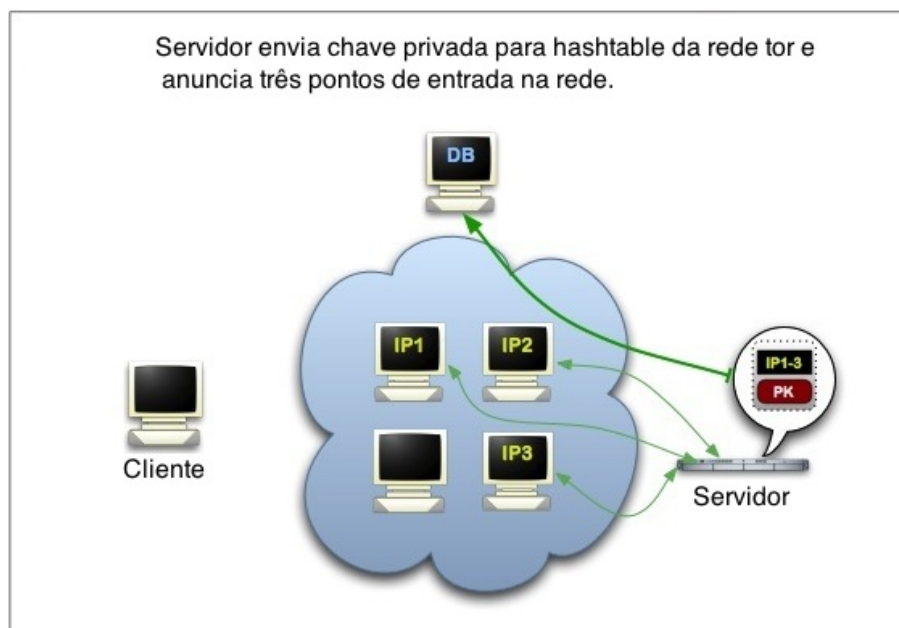


Figura 2. (<https://www.torproject.org/docs/hidden-services.html.en>)

Esses pontos de entrada apenas informam a chave pública de criptografia para o acesso ao servidor não informando o ip.

A partir deste momento o *Hidden Service* constrói um "descriptor de serviços ocultos" que contém a descrição dos pontos de entrada e a chave pública e faz o *upload* para uma *hashtable* distribuída pela rede TOR. O descriptor será encontrado na *hashtable* distribuída pela rede, pelo *host* que tiver o endereço do servidor. Após ser feito o *download* do descriptor o *host* cliente então já sabe os pontos de entrada do servidor e tem a chave pública de criptografia para estabelecer a conexão. O cliente então escolhe um *relay* aleatório para ser o ponto de encontro entre cliente e servidor. Então o cliente manda uma mensagem para um ponto de entrada do servidor, pedindo para ser entregue para o *hidden service*.

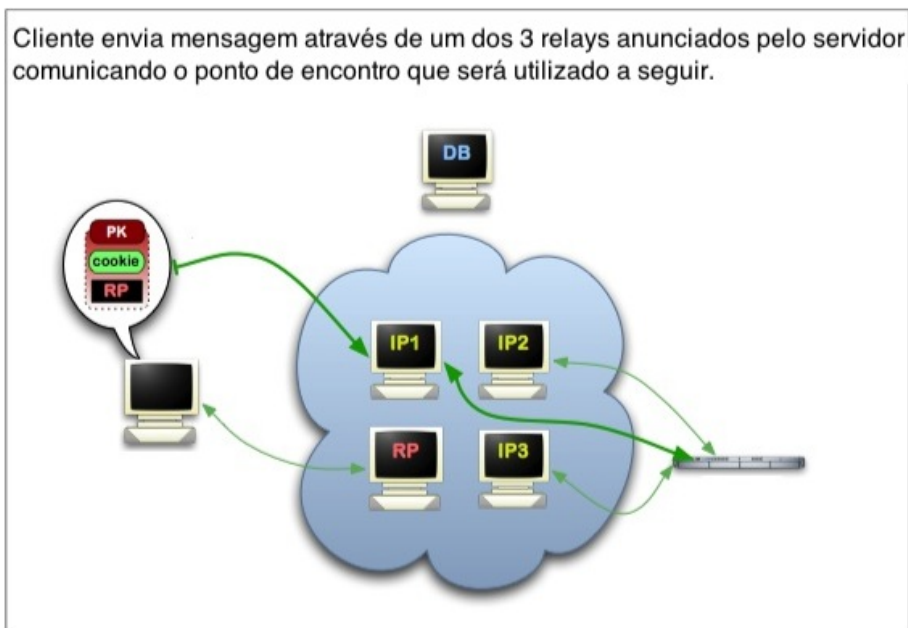


Figura 3. (<https://www.torproject.org/docs/hidden-services.html.en>)

O servidor descriptografa a mensagem e encontra a informação sobre o ponto de encontro(RP). Então é criado um circuito até o ponto de encontro(RP). É importante que o *hidden service* mantenha os mesmo 3 pontos de entrada configurados inicialmente para evitar um ataque via *relay* contaminado caso caia um dos *relays*. Por fim o ponto de entrada informa o cliente que o servidor estabeleceu uma conexão com sucesso. A partir deste momento cliente e servidor usam seus circuitos de entrada para trocarem mensagens através do ponto de encontro.

Cliente e servidor se comunicando através do ponto de encontro.

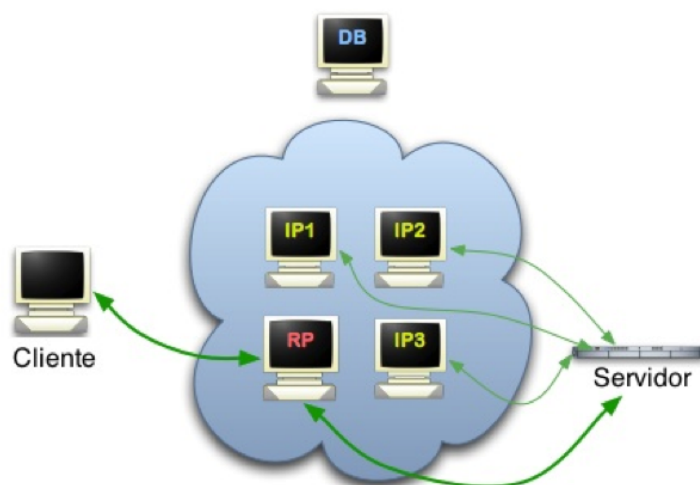


Figura 4. (<https://www.torproject.org/docs/hidden-services.html.en>)

### 3. Vidalia

Vidalia é um software disponibilizado pelo projeto TOR, que tem por função se tornar um *relay* de saída ou não-saída. No modo não saída, apenas conexões destinadas a estabelecer circuitos dentro da rede são criadas. No modo saída quando algum usuário solicita acesso a um serviço fora da rede TOR ele utiliza a conexão de um *relay* de saída. Dessa forma qualquer tipo de tráfego pode sair pelo *relay*, e o ip que fica registrado em uma conexão com um servidor http por exemplo é o do *relay* de saída, mantendo o usuário da rede TOR anônimo, mas expondo o ip do *relay*.

E aqui a janela onde o Vidalia é setado para se tornar um *relay* de saída.

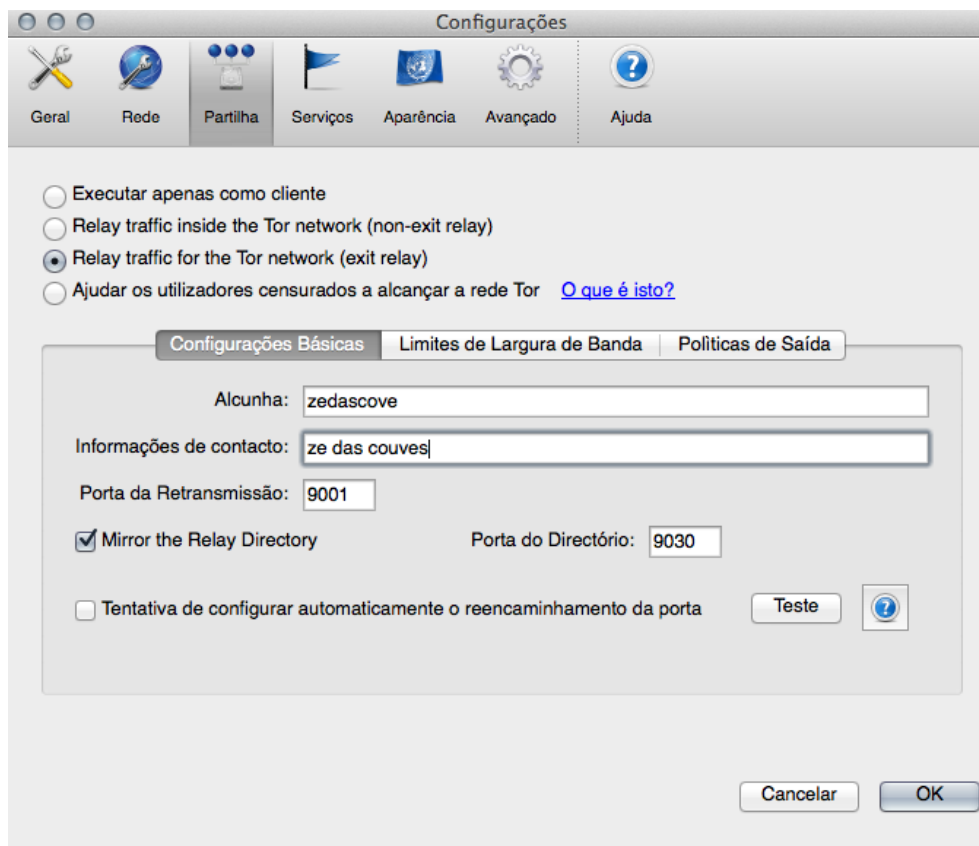


Figura 5.

## Menu de configurações

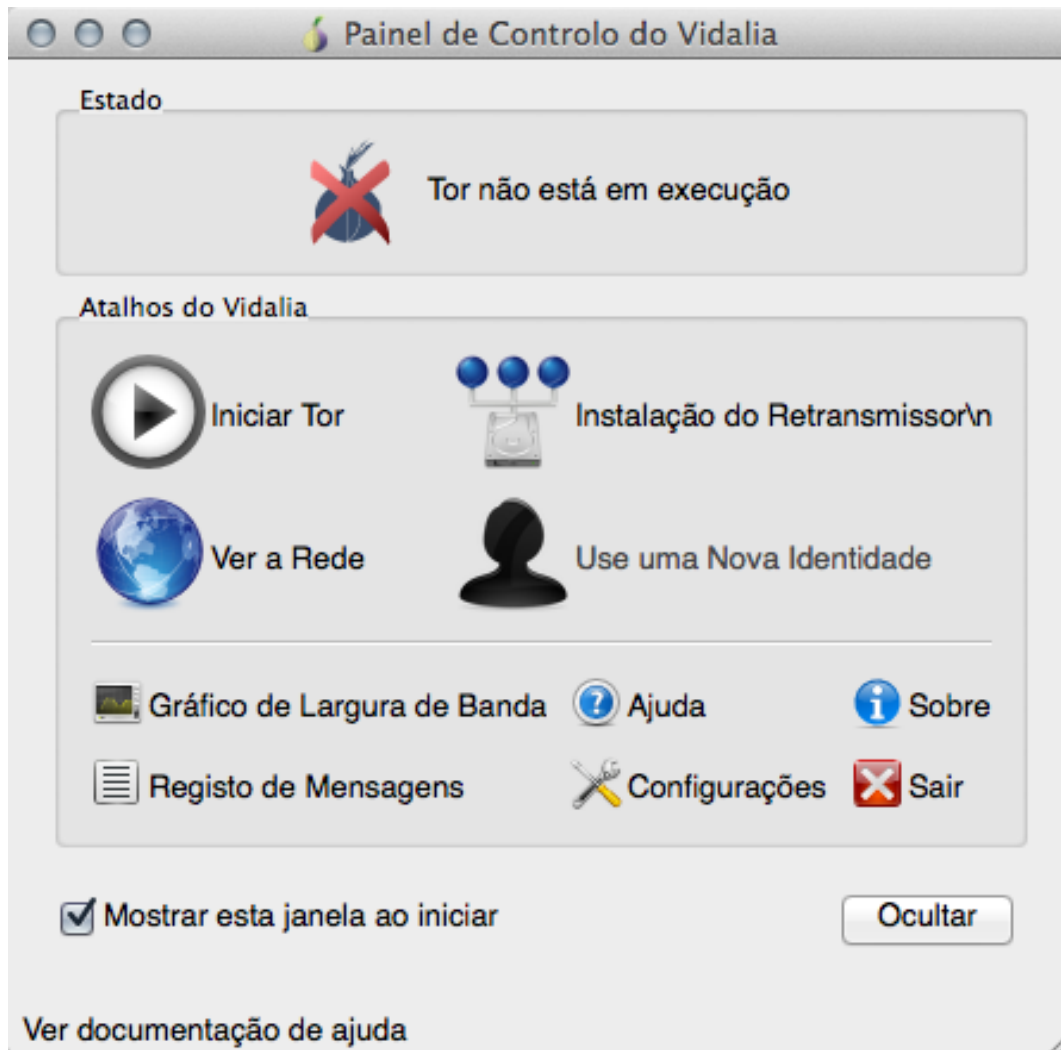


Figura 6.

### 3.1. Ambiente de testes

Os testes foram realizados em um notebook Asus N43SN com processador I7 e 6GB de RAM, conectado via wifi em um modem da operadora GVT em um link de 35Mb/s de *download* e 3Mb/s de upload. Foram encerrados todos os processos que poderiam gerar algum tráfego ficando apenas serviços do sistema operacional e o software Vidalia utilizando a rede. Os pacotes foram capturados no wireshark e filtrados os serviços que trocam mensagens via rede em texto plano que pode ser visualizado facilmente na captura.

### 3.2. Primeira captura de pacotes

Foram capturados pacotes durante aproximadamente 2h ininterruptas. Os filtros utilizados no wireshark foram smtp, pop, dns, telnet e http. Porém nem todos protocolos mostraram resultados na captura, dessa forma serão mostrados apenas os que tiveram resultados positivos na busca.

### 3.2.1. Primeira filtragem de protocolo *HTTP*

Nesta primeira filtragem mostrada na figura 7 é exibida uma captura usando a opção "follow tcp stream" do *Wireshark* em que mostra uma requisição *HTTP* em um site que aponta o ip do *relay* de saída.

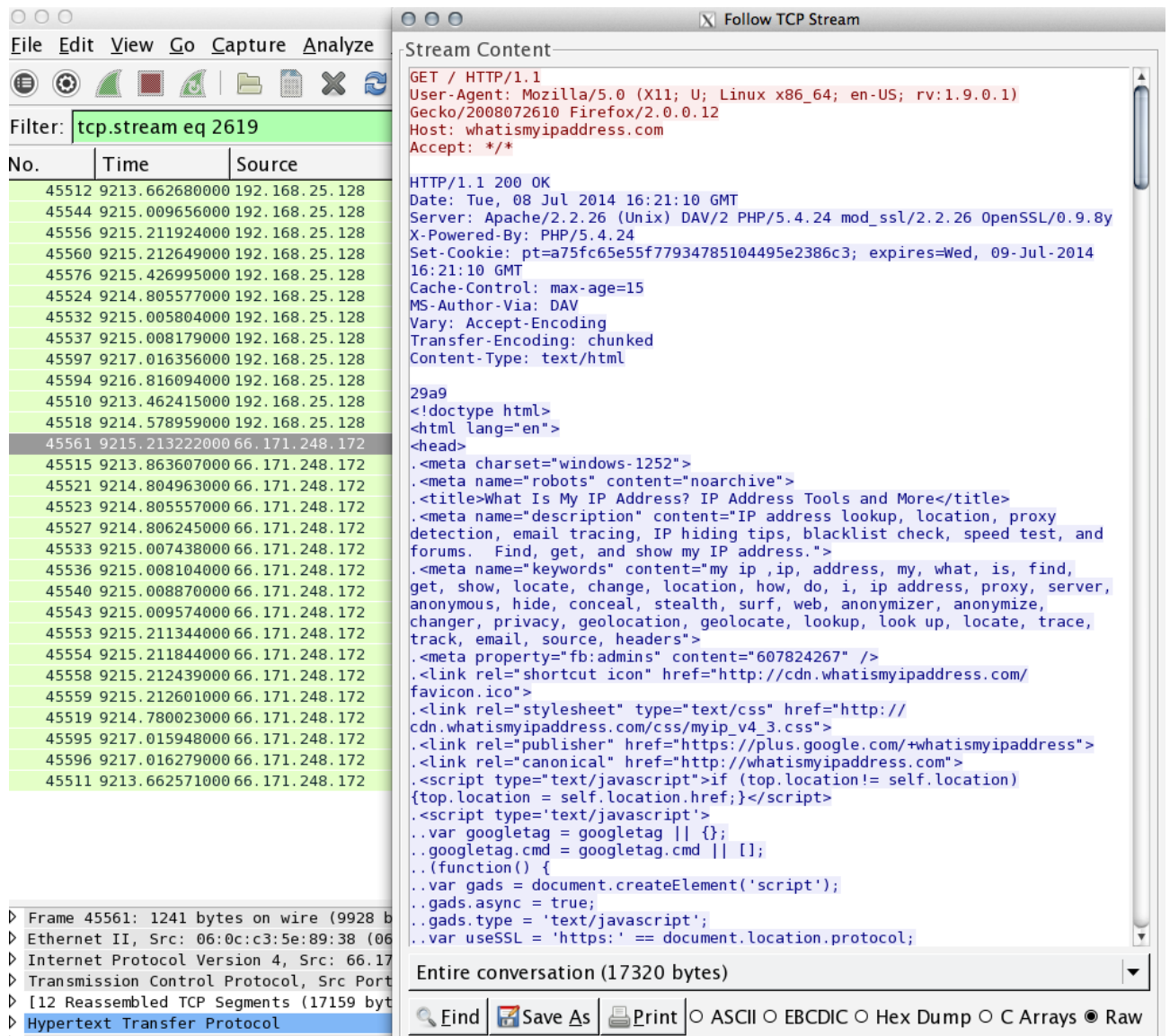


Figura 7.

### 3.2.2. Segunda filtragem de de protocolo *HTTP*

Nesta segunda filtragem mostrada na figura 8 vemos uma captura usando a opção "follow tcp stream" do *Wireshark* em que é exibida uma chave pública de certificado sendo recebida.

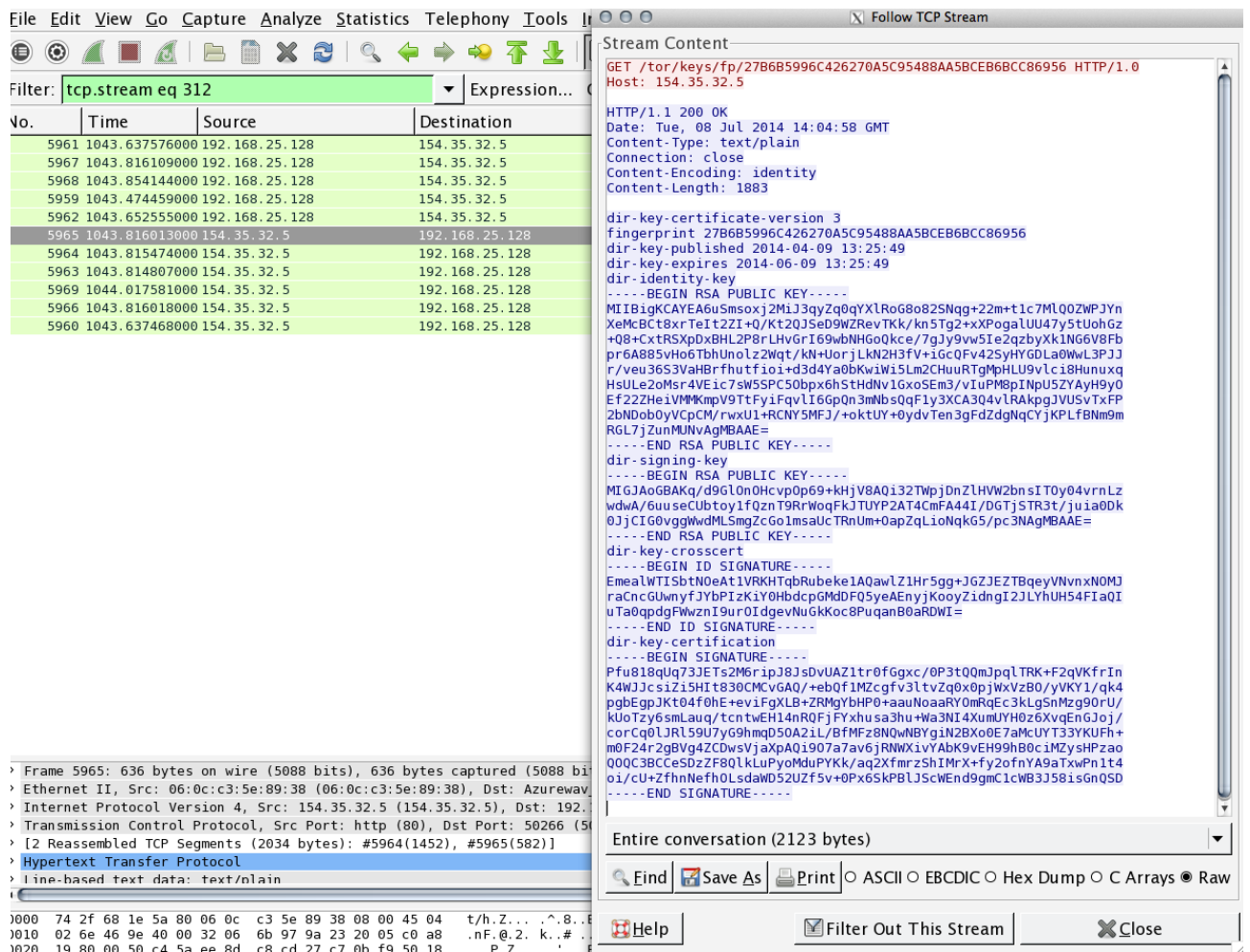


Figura 8.

### 3.2.3. Terceira filtragem de de protocolo *HTTP*

Nesta terceira filtragem mostrada na figura 9 vemos uma captura usando a opção "follow tcp stream" do *Wireshark* em que mostra uma requisição *HTTP* a um site que localiza o a cidade onde está o *relay* de saída em teste.



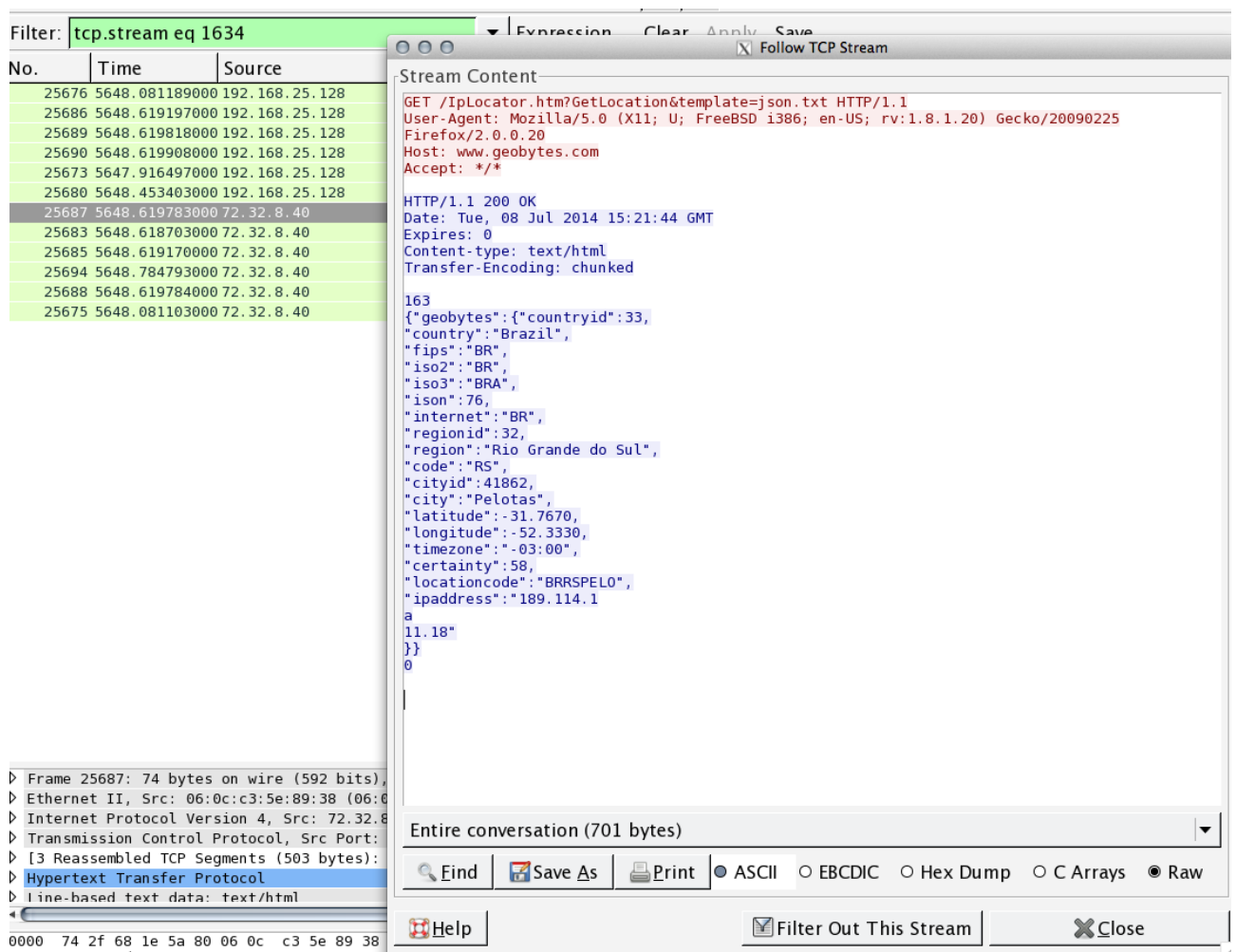


Figura 9.

### 3.2.4. Filtragem de de protocolo DNS

Nesta segunda filtragem mostrada na figura 10 vemos uma captura de requisição *DNS* que mostra as requisições feitas pelos outros *relays*.

| Info   |
|--|
| Standard query response 0x7371 No such name                        |
| Standard query response 0xc04f No such name                        |
| Standard query 0xd69b A WwW.GoogLE.com                             |
| Standard query 0x1df5 AAAA WwW.goOgLE.Com                          |
| Standard query 0x1df6 A WwW.Mit.EDU                                |
| Standard query 0x7a15 AAAA wWw.Mit.eDU                             |
| Standard query 0xd971 A www.YahOo.CoM                              |
| Standard query 0xd396 AAAA WwW.YahOo.Com                           |
| Standard query 0x99d7 A WwW.SLaSHdot.org                           |
| Standard query 0xc210 AAAA wwW.sLASHDOT.ORG                        |
| Standard query response 0xd69b A 74.125.225.18 A 74.125.225.20 A   |
| Standard query response 0x1df5 AAAA 2607:f8b0:4009:807::1010       |
| Standard query response 0xd971 CNAME fd-fp3.wg1.b.YahOo.CoM CNAME  |
| Standard query response 0xd396 CNAME fd-fp3.wg1.b.YahOo.Com CNAME  |
| Standard query response 0x7a15 CNAME www.mit.edu.edgekey.net CNAME |
| Standard query response 0x99d7 A 216.34.181.48                     |
| Standard query response 0xc210                                     |
| Standard query response 0x1df6 CNAME www.mit.edu.edgekey.net CNAME |

Figura 10.

### 3.3. Segunda captura de pacotes

Foram capturados pacotes durante aproximadamente 4h ininterruptas. Os filtros utilizados no *Wireshark* foram *smtp*, *pop*, *dns*, *telnet* e *http*. Porém nem todos protocolos mostraram resultados na captura, dessa forma serão mostrados apenas os que tiveram resultados positivos na busca.

#### 3.3.1. Primeira filtragem de protocolo POP

Nesta primeira filtragem mostrada na figura 11 vemos uma captura que mostra uma conexão estabelecida com um servidor de email POP utilizando o *relay* de saída.

| Source         | Destination    | Protocol | Length | Info   |
|----------------|----------------|----------|--------|--|
| 192.168.25.128 | 38.229.72.22   | POP      | 208    | C: POST / HTTP/1.0                                     |
| 38.229.72.22   | 192.168.25.128 | IMF      | 100    | [Malformed Packet]                                     |
| 192.168.0.24   | 81.173.240.81  | POP      | 298    | C: \026\003\001\000\357\001\000\000\353\003\003"\26;   |
| 81.173.240.81  | 192.168.0.24   | IMF      | 815    | \026\003\003\000>\002\000\000:\003\0035\274<\244 , \   |
| 192.168.0.24   | 81.173.240.81  | POP      | 180    | C: \026\003\003\000F\020\000\000BA\004h(%\257Q\377\2   |
| 81.173.240.81  | 192.168.0.24   | IMF      | 105    | \024\003\003\000\001\001\026\003\003\000(\317\205\02   |
| 192.168.0.24   | 81.173.240.81  | POP      | 92     | C: \027\003\003\000!q\003\362\034C\362\026\350_e\351   |
| 81.173.240.81  | 192.168.0.24   | IMF      | 1466   | \027\003\003\005\364\317\205\022\364\377;\241#\306     |
| 81.173.240.81  | 192.168.0.24   | IMF      | 171    | \344\307\241C hclW\265oy\303\2727\357\350D\032\357' ]C |
| 192.168.0.24   | 81.173.240.81  | POP      | 1466   | C: \027\003\003\364\317\205\022\364\377;\241#\306      |
| 192.168.0.24   | 81.173.240.81  | POP      | 447    | C: \231L\204&\3221C\220\220\332\033I4\241\211n\020\6   |
| 192.168.0.24   | 81.173.240.81  | POP      | 597    | C: \027\003\003\002\032q\003\362\034C\362\026\352\24   |
| 81.173.240.81  | 192.168.0.24   | IMF      | 597    | \027\003\003\002\032\317\205\022\364\377;\2421JA\2     |
| 192.168.0.24   | 81.173.240.81  | POP      | 597    | C: \027\003\003\002\032q\003\362\034C\362\026\353D\3   |
| 81.173.240.81  | 192.168.0.24   | IMF      | 597    | \027\003\003\002\032\317\205\022\364\377;\243\260\     |
| 192.168.0.24   | 81.173.240.81  | POP      | 597    | C: \027\003\003\002\032q\003\362\034C\362\026\354\3f-  |
| 81.173.240.81  | 192.168.0.24   | IMF      | 597    | \027\003\003\002\032\317\205\022\364\377;\244\364\     |

Figura 11.

### 4. Wiki e mecanismos de busca

O site mais difundido e recomendado para começar a navegar na *Deep Web* é a *Hidden Wiki*. Lá são encontrados alguns links iniciais de vários tipos de conteúdos lícitos e ilícitos, porém ele muda com certa frequência e sempre há mirrors replicando o conteúdo. A *Hidden Wiki* geralmente é a porta de entrada para os primeiros acessos

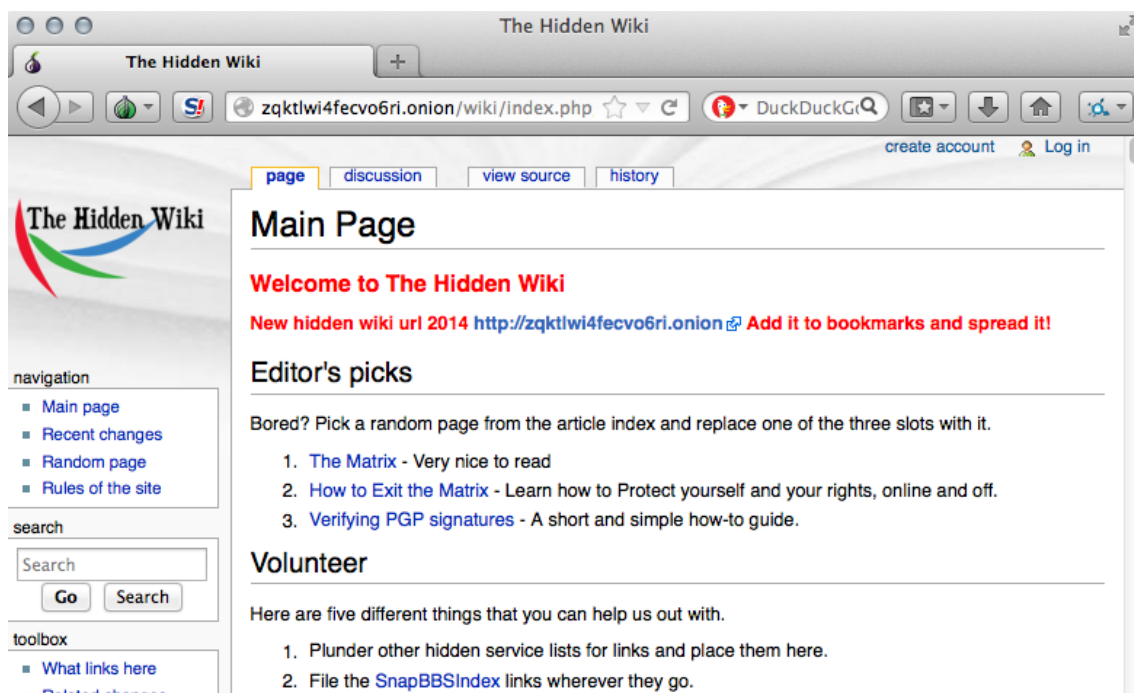


Figura 12.

## 5. Segurança

Muito é debatido sobre segurança na *Deep Web*. O anonimato garantido pela rede *TOR* acaba se tornando refúgio de usuários com interesses ilícitos. Isso torna a rede *TOR* um tanto perigosa para os desavisados.

Todos os usuários da rede *TOR* devem se proteger na *Deep Web*. O ideal é utilizar uma máquina virtual hospedada no próprio PC do usuário que só deve ser utilizada em modo *NAT* no *VMWare* e ligada apenas para acessar a *Deep Web*.

Enfim todo cuidado é pouco. Evitar fazer *downloads* de arquivos e nunca dar informações pessoais. Mantenha-se anônimo. Quanto a se tornar um *relay* da rede *TOR* tenha em mente que irão circular dados de qualquer tipo pelo seu *relay*. Se for um relay de saída pior ainda pois pode ser que o acesso a algum conteúdo ilícito fique registrado como tendo saído do seu IP. Não confie em ninguém dentro do *TOR Browser*.

## 6. Conclusões

A *Deep Web* tem uma importante função na internet de hoje que é promover o anonimato dos usuários quando necessário. Apesar da má fama por conta dos usuários inescrupulosos, a tecnologia utilizada é bastante eficiente e existem diversos projetos atrelados a rede *TOR*, como *browsers* para todos sistemas operacionais, smartphones e live cd com sistema operacional dedicado.

Para ter um *TOR relay* e poder contribuir para a rede *TOR* fechar seus circuitos criptografados basta ter um link e configurar as portas de entrada caso se use uma rede com *NAT*. Os conteúdos encontrados na maioria das vezes nunca foi disponibilizado na internet comum(*surface*), o que torna a navegação na *Deep Web* uma atividade interessante.

Por questão de segurança em países que mantêm a internet comum sob forte vigilância a *Deep Web* é de utilidade extrema. Já que impossibilita que as mensagens sejam filtradas, dessa forma podendo até salvar vidas em regimes de governo mais autoritários. Porém ao usuário comum recomenda-se toda cautela, já que lá os criminosos estão anônimos e protegidos pelo anonimato da rede. Não devemos condenar a rede *TOR* como um todo já que a internet comum também possui seu lado negro com *cybercrimes* bem conhecidos como vazamento de dados trocados por e-mail.

Baseado nas capturas feitas pelo *Wireshark* se conclui também que apesar da rede *TOR* manter o anonimato do usuário, o conteúdo acessado através de protocolos que usam *plain-text* na formação do pacote pode ser visualizado pelo administrador do relay de saída. De forma que se o objetivo é tornar oculto para qualquer relay os dados trafegados é necessário trocar dados apenas dentro da *Deep Web* e nunca usar um relay de saída.

Os *relays* são espalhados por toda internet, utilizando rotas não otimizadas entre cada relay que fecha os circuitos. Isso aumenta consideravelmente o atraso na troca de pacotes entre os relays tornando lento o acesso a *Deep Web*.

## 7. Referências

Artigo ?*What Are TOR Hidden Services?*(2012). Disponível em:  
<http://www.infosecisland.com/blogview/21635-What-Are-ToR-Hidden-Services.html>.  
Acesso em: 01/07/2014

Documentação do rede TOR (2014). Disponível em:  
<https://www.torproject.org/docs/documentation>. Acesso em: 23/03/2014.

Documentação do Hidden Service Protocol (2014). Disponível em:  
<https://www.torproject.org/docs/tor-hidden-service.html.en>. Acesso em: 23/03/2014

Domain Names Service (1987). Disponível em:  
<http://www.ietf.org/rfc/rfc1035.txt>. Acesso em 09/07/2014

Estatísticas da rede TOR (2014). Disponível em:  
<https://metrics.torproject.org/users.html>. Acesso em 23/03/2014

Hypertext Transfer Protocol (1999). Disponível em:  
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>. Acesso em: 09/07/2014

Post Office Protocol (1996). Disponível em: <http://www.ietf.org/rfc/rfc1939.txt>  
Acesso em: 09/07/2014

Hypertext Transfer Protocol (1999). Disponível em:  
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>. Acesso em: 09/07/2014

Wireshark Documentation. Disponível em: <http://www.wireshark.org/docs/>.  
Acesso em: 09/07/2014