

Entendendo a *Deep Web*

Simei Tabordes Gonçalves

¹Faculdade de Tecnologia Senac Pelotas
Rua Gonçalves Chaves, 602 - Pelotas - RS - Brasil - Caixa Postal - 96.015-560
Curso Superior de Tecnologia em Redes de Computadores

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 foi o termo criado para descrever os conteúdos que só pode ser acessados através da rede TOR(*The Onion Network*)[TOR 2014], 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. 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 mostrar 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

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. O objetivo maior é tornar anônimo o endereço 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.

A figura 1 mostra o número de usuários conectados nos últimos 3 meses. O número chegou a 3.000.000 em abril e se mantém na média de 2.300.000. Em [TOR Metrics 2014] <https://metrics.torproject.org/users.html> há estatísticas atualizadas sobre a *Deep Web*.

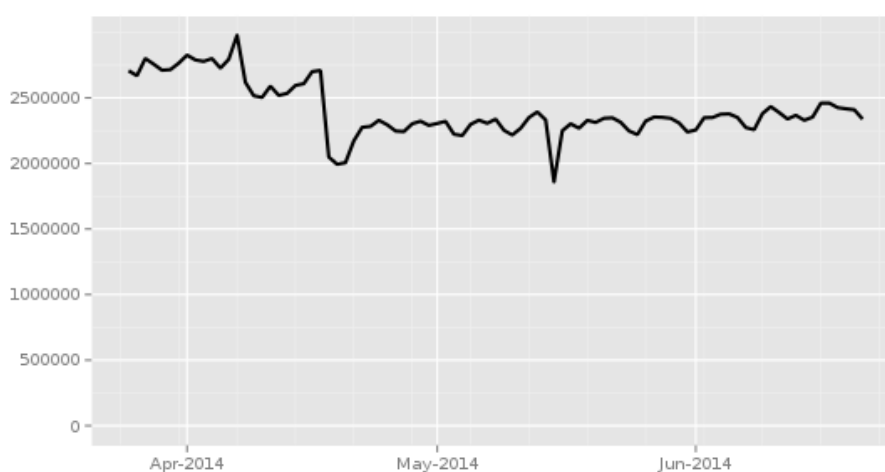


Figura 1. Número de usuários conectados nos últimos 3 meses.

2.1. TOR Browser

É o *browser* oficial da rede TOR. É uma versão modificada do conhecido Firefox da Mozilla [Firefox 2014]. Roda em mídias removíveis e possui versões para diversos sistemas operacionais. 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 e 6999.

3. 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.

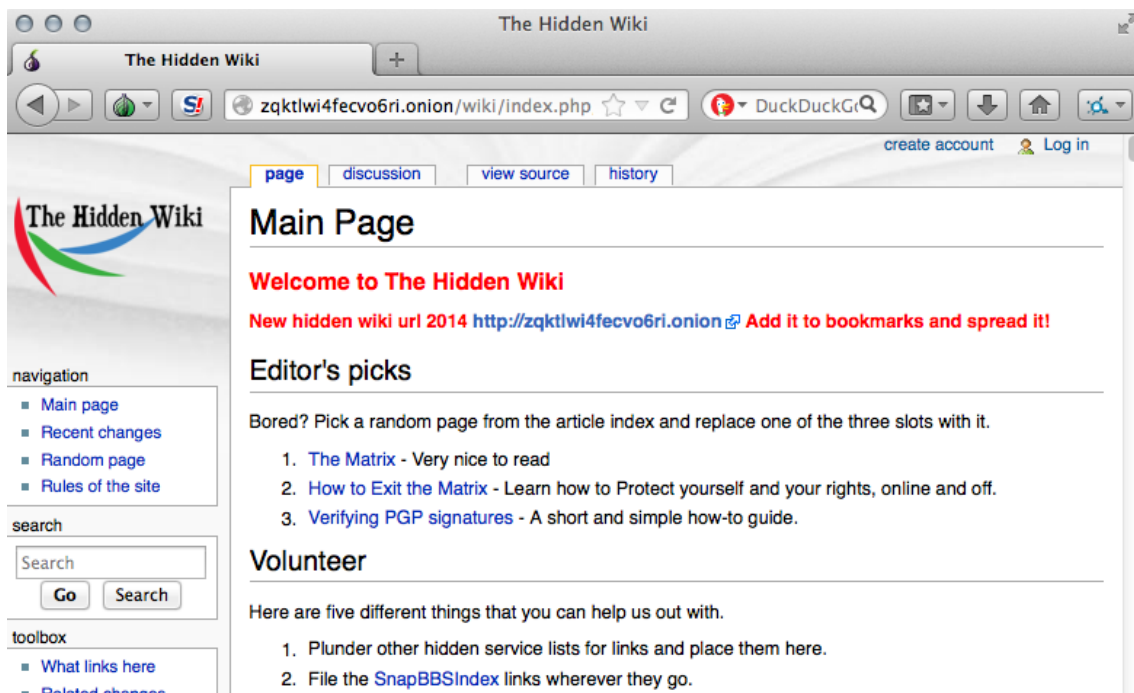


Figura 2. Página inicial para maioria dos usuários. *The Hidden Wiki*.

3.1. *Hidden Service Protocol*

O *HSP* [Hidden Service Protocol 2014] 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. Esses pontos de entrada apenas informam a chave pública de criptografia para o acesso ao servidor não informando o ip. Esse processo é ilustrado na figura 3.

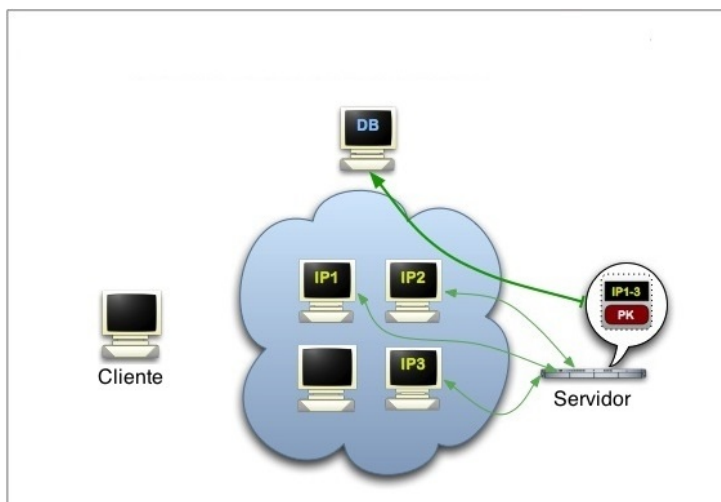


Figura 3. O computador DB contém a *Hashtable* distribuída que é replicada entre os *relays*. Primeira etapa da criação do circuito.

A partir deste momento o *Hidden Service* constrói um descritor 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 descritor será encontrado na *hashtable* [Hashtable 2011] distribuída pela rede, pelo *host* que tiver o endereço do servidor. Após ser feito o *download* do descritor 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 escolhe um *relay* aleatório para ser o ponto de encontro entre cliente e servidor e manda uma mensagem para um ponto de entrada do servidor, pedindo para ser entregue para o *hidden service*. Esse processo é ilustrado na figura 4.

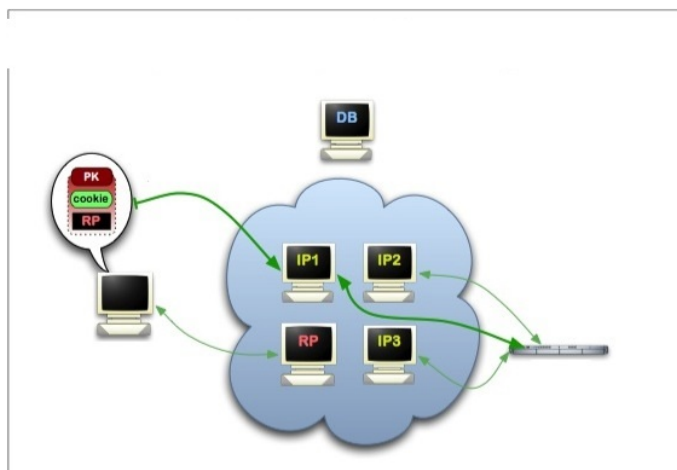


Figura 4. Segunda etapa da criação do circuito.

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. 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. Esse processo é ilustrado na figura 5.

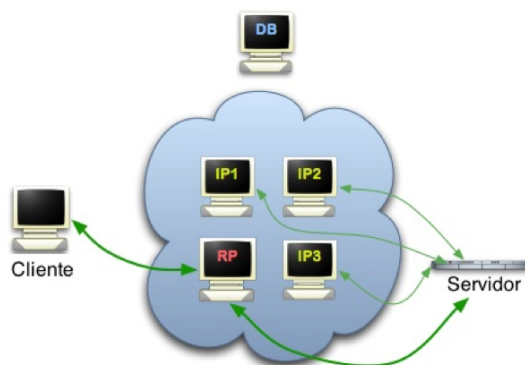


Figura 5. Terceira etapa da criação do circuito.

4. 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*. Na figura 6 vemos a janela que mostra a configuração de *exit relay*, ou *non-exit relay* [HTTP 1999].

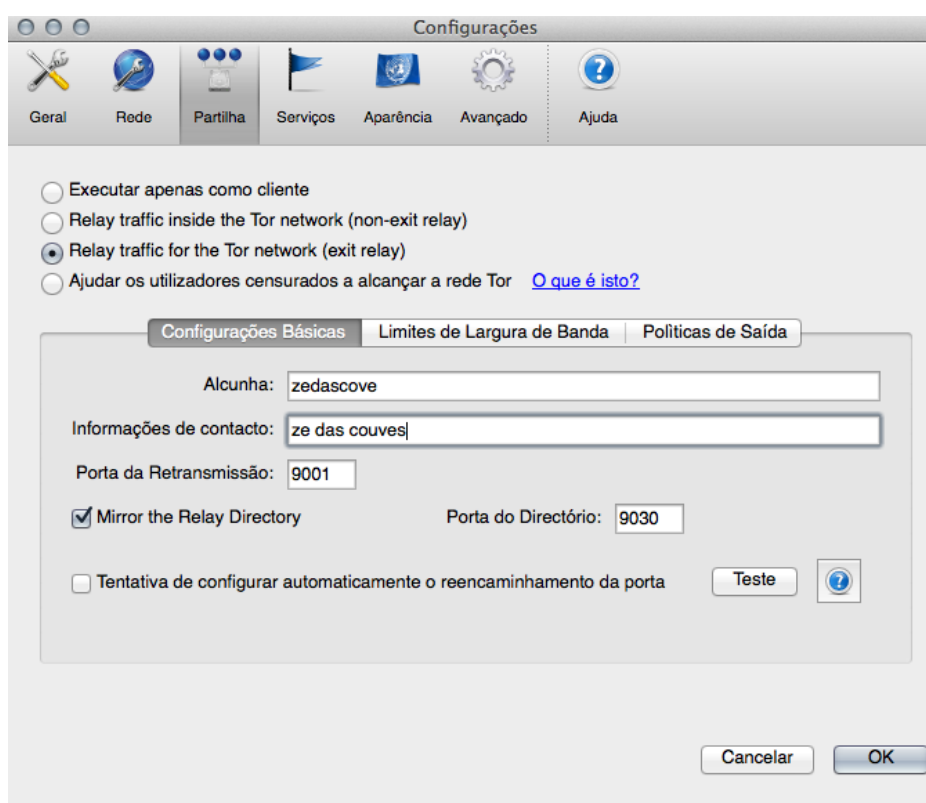


Figura 6. E aqui a janela onde o Vidalia é setado para se tornar um *relay* de saída.

4.1. Ambiente de testes

Os testes foram realizados em um notebook Asus N43SN com processador I7 e 6GB de RAM, conectado 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 [Wireshark 2014] e remontados no [Network Miner 2014].

4.2. Primeira captura de pacotes

Foram capturados pacotes durante aproximadamente 2h ininterruptas. Os filtros utilizados no *Wireshark* pop [POP 1996], dns [DNS 1987] e http [HTTP 1999].

4.2.1. 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. É mostrado também o filtro utilizado no *Wireshark* para abrir o *stream* no campo *Filter*. No campo *Stream Content* na 4ª linha é mostrado o endereço do site que está sendo acessado.

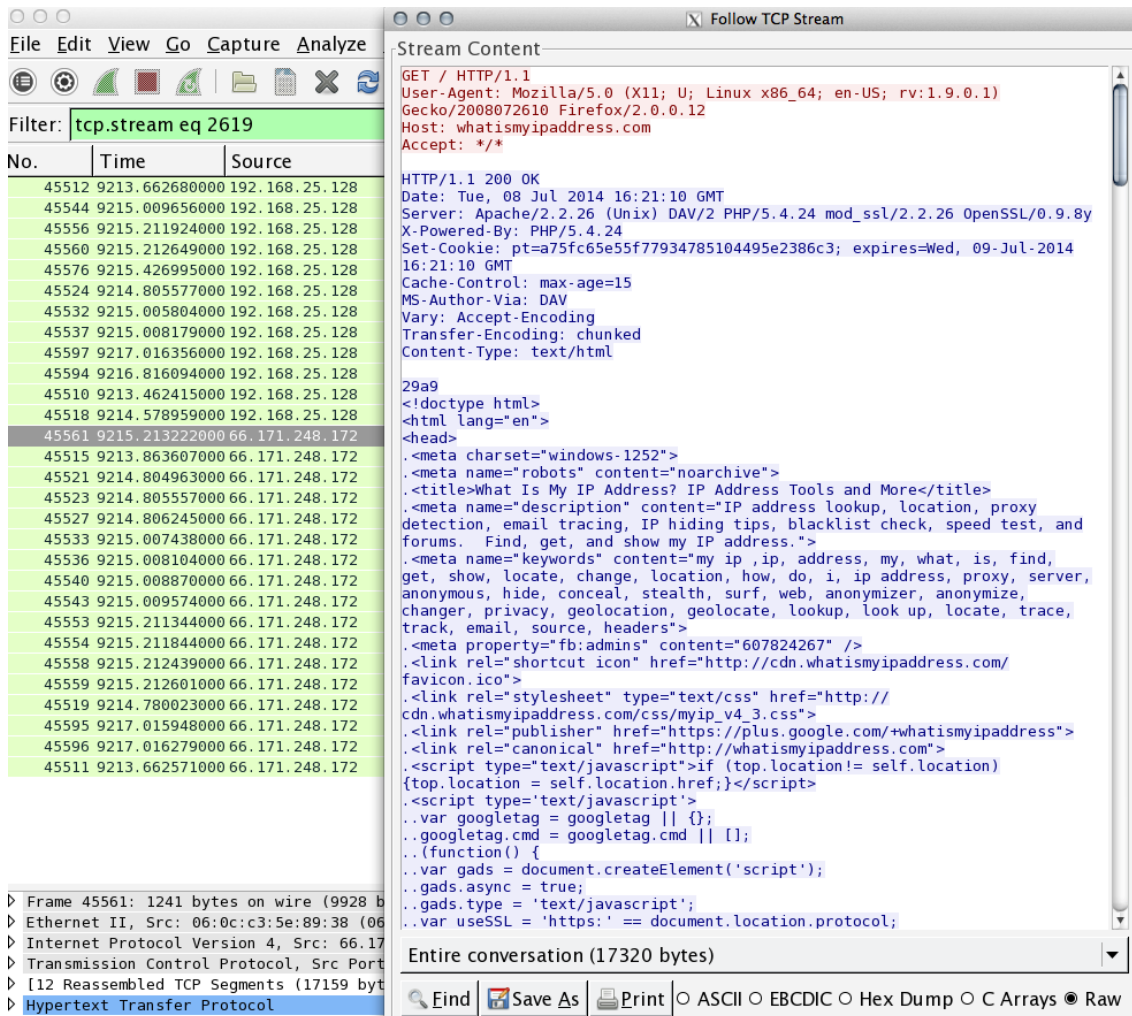


Figura 7. Screenshot do Wireshark mostrando o conteúdo do pacote.

4.2.2. Segunda filtragem de protocolo HTTP

Nesta segunda filtragem mostrada na figura 8 é exibida uma captura usando a opção "follow tcp stream" do *Wireshark* em que é exibida uma chave pública de certificado sendo recebida. No campo *Stream Content* é mostrado também na primeira linha que a chave trocada está sendo utilizada pela rede TOR.

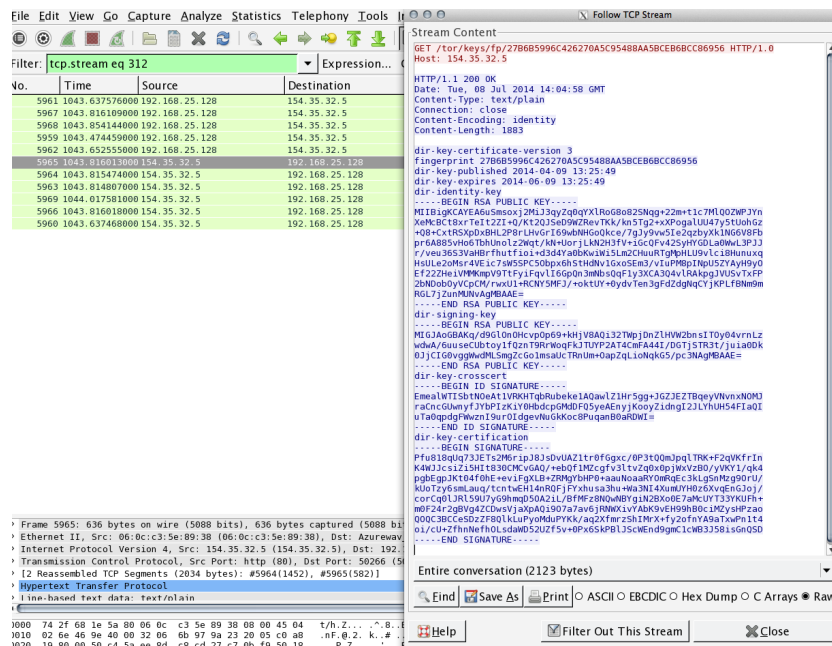


Figura 8. Chave sendo trocada dentro da rede TOR.

4.2.3. Terceira filtragem de de protocolo *HTTP*

Nesta terceira filtragem mostrada na figura 9 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. No campo *Stream Content* da figura 9 na primeira e quarta linha o endereço do site e abaixo os dados enviados ao cliente pelo servidor *http*.

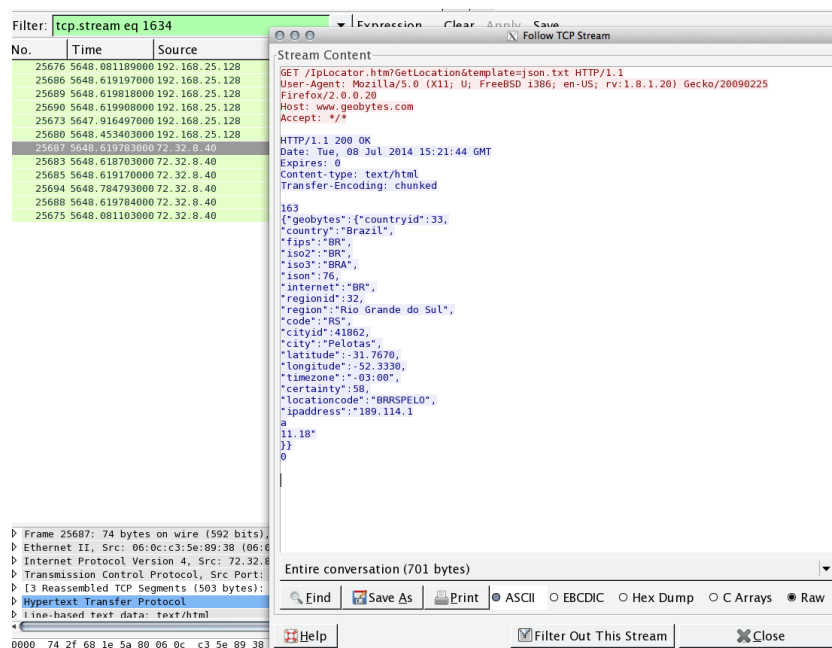


Figura 9. Tráfego *HTTP* capturado no *Wireshark*.

4.2.4. Filtragem de de protocolo DNS

Nesta segunda filtragem foi feita a captura de pacotes de requisições DNS feitas pelo *relay* originada pelos usuários da rede TOR saindo para a *Web* comum. A captura é mostrada na figura 10.

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.go0GLE.Com
Standard query 0x1df6 A WwW.Mit.EDU
Standard query 0x7a15 AAAA wWw.Mit.eDU
Standard query 0xd971 A www.Yah00.CoM
Standard query 0xd396 AAAA Www.YaH0o.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.Yah00.CoM CNAME
Standard query response 0xd396 CNAME fd-fp3.wg1.b.YaH0o.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. Requisições ao sites conhecidos como *www.google.com* sendo capturadas.

4.3. Segunda captura de pacotes

Foram capturados pacotes durante aproximadamente 4h ininterruptas.

4.3.1. Primeira filtragem de protocolo POP

Nesta primeira filtragem mostrada na figura 11 é exibida 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(\%2570\377\;
81.173.240.81	192.168.0.24	IMF	105	\024\003\003\000\001\001\026\003\003\000(\317\205\0;
192.168.0.24	81.173.240.81	POP	92	C: \027\003\003\000!q\003\362\034C\362\026\350_e\35;
81.173.240.81	192.168.0.24	IMF	1466	\027\003\003\005\364\317\205\022\022\0364\377;\241#\30f
81.173.240.81	192.168.0.24	IMF	171	\344\307\241C_hcW\265oy\303\2727\357\3500\032\357\1C
192.168.0.24	81.173.240.81	POP	1466	C: \027\003\003\003\003\003\362\034C\362\026\351\277xb\
192.168.0.24	81.173.240.81	POP	447	C: \2311\2046\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\2;
81.173.240.81	192.168.0.24	IMF	597	\027\003\003\002\032\317\205\022\022\0364\377;\2421JA\;
192.168.0.24	81.173.240.81	POP	597	C: \027\003\003\002\032q\003\362\034C\362\026\3530\;
81.173.240.81	192.168.0.24	IMF	597	\027\003\003\002\032\317\205\022\022\0364\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\6-
81.173.240.81	192.168.0.24	IMF	597	\027\003\003\002\032\317\205\022\022\0364\377;\244\364\

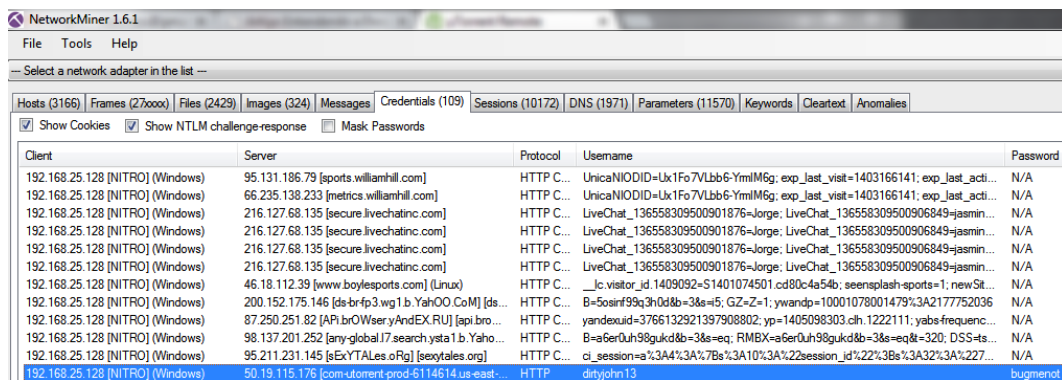
Figura 11. Tráfego entre *hosts* utilizando o protocolo POP para recebimento de *e-mail*.

4.4. Análise de uma captura de pacotes com o programa *Network Miner 1.6.1*

O *NetworkMiner* é uma ferramenta de análise de pacotes para sistemas operacionais *Microsoft* que podem detectar o sistema operacional e portas abertas em *hosts*. Os pacotes da segunda captura de pacotes no *Wireshark* foram salvos com a extensão *.pcap* para então ser aberto o arquivo no *NetworkMiner*. O *NetworkMiner* também pode extrair arquivos transmitidos nos pacotes capturados.

4.4.1. Filtragem de *credentials*

Nesta filtragem são mostrados usuários e senhas capturados nos pacotes filtrados. Na figura 12 é mostrado o usuário e senha de acesso a um *host*.

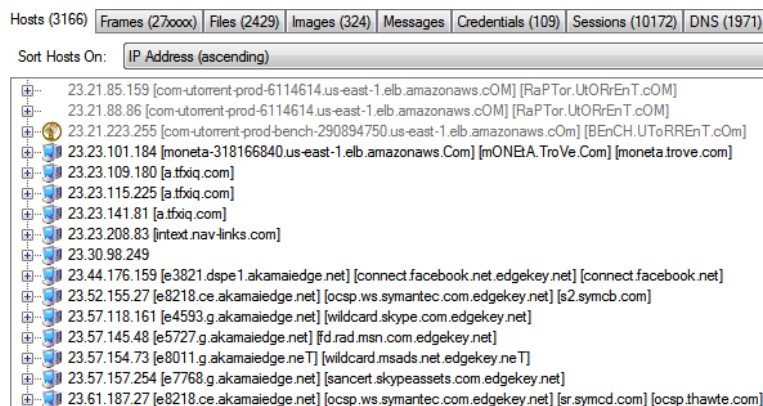


Client	Server	Protocol	Username	Password
192.168.25.128 [NITRO] (Windows)	95.131.186.79 [sports.williamhill.com]	HTTP C...	UnicaNIODID=Ux1Fo7Vlbb6-YmlMfg; exp_last_vist=1403166141; exp_last_acti...	N/A
192.168.25.128 [NITRO] (Windows)	66.235.138.233 [metrics.williamhill.com]	HTTP C...	UnicaNIODID=Ux1Fo7Vlbb6-YmlMfg; exp_last_vist=1403166141; exp_last_acti...	N/A
192.168.25.128 [NITRO] (Windows)	216.127.68.135 [secure.livechatinc.com]	HTTP C...	LiveChat_136558309500901876-Jorge; LiveChat_136558309500906849-jasmin...	N/A
192.168.25.128 [NITRO] (Windows)	216.127.68.135 [secure.livechatinc.com]	HTTP C...	LiveChat_136558309500901876-Jorge; LiveChat_136558309500906849-jasmin...	N/A
192.168.25.128 [NITRO] (Windows)	216.127.68.135 [secure.livechatinc.com]	HTTP C...	LiveChat_136558309500901876-Jorge; LiveChat_136558309500906849-jasmin...	N/A
192.168.25.128 [NITRO] (Windows)	216.127.68.135 [secure.livechatinc.com]	HTTP C...	LiveChat_136558309500901876-Jorge; LiveChat_136558309500906849-jasmin...	N/A
192.168.25.128 [NITRO] (Windows)	46.18.112.39 [www.boylesports.com] (Linux)	HTTP C...	__jc_visitor_id.1409092-S1401074501.cd80c4a54b; seensplash-sports=1; newSt...	N/A
192.168.25.128 [NITRO] (Windows)	200.152.175.146 [ds-br-fp-3.wg1.b.Yahoo.CoM] [ds...	HTTP C...	B=5osinf99q3h0d&b=3&s=e5; GZ=Z=1; ywandp=10001078001479%3A2177752036	N/A
192.168.25.128 [NITRO] (Windows)	87.250.251.82 [API.brOWser.yAndEX.RU] [api bro...	HTTP C...	yandexuid=3766132921397908802; yp=1405098303.ch.1222111; yabs.freque...	N/A
192.168.25.128 [NITRO] (Windows)	98.137.201.252 [any-global17.search.ysta1.b.Yaho...	HTTP C...	B=6er0uh98gukd&b=3&s=eq; RMBX=a6er0uh98gukd&b=3&s=eq&t=320; DSS=ts...	N/A
192.168.25.128 [NITRO] (Windows)	95.211.231.145 [sExYTAles.oRg] [sexytales.org]	HTTP C...	cl_session=a%3A4%3A%7Bs%3A10%3A%22session_id%22%3B%3A32%3A%227...	N/A
192.168.25.128 [NITRO] (Windows)	50.19.115.176 [com-torrent-prod-6114614.us-east-...	HTTP	dirtyjohn13	bugmenot

Figura 12. Usuário e senha capturados e visualizados pelo *NetworkMiner*.

4.4.2. Filtragem de *hosts*

Nesta outra amostra da filtragem mostrada na figura 13 é possível visualizar os *hosts* acessados pelos usuários através da rede TOR.



Hosts (3166)	Frames (27xxxx)	Files (2429)	Images (324)	Messages	Credentials (109)	Sessions (10172)	DNS (1971)
Sort Hosts On: IP Address (ascending)							
23.21.85.159	[com-torrent-prod-6114614.us-east-1.elb.amazonaws.com]	[RaPTor.UtORrEnT.cOm]					
23.21.88.86	[com-torrent-prod-6114614.us-east-1.elb.amazonaws.com]	[RaPTor.UtORrEnT.cOm]					
23.21.223.255	[com-torrent-prod-bench-290894750.us-east-1.elb.amazonaws.com]	[BEnCH.UToRREnT.cOm]					
23.23.101.184	[moneta-318166840.us-east-1.elb.amazonaws.com]	[mONETA.TroVe.Com]					
23.23.109.180	[a.tfxiq.com]						
23.23.115.225	[a.tfxiq.com]						
23.23.141.81	[a.tfxiq.com]						
23.23.208.83	[rntext.nav-links.com]						
23.30.98.249							
23.44.176.159	[e3821.dspe1.akamaiedge.net]	[connect.facebook.net.edgekey.net]					
23.52.155.27	[e8218.ce.akamaiedge.net]	[ocsp.ws.symantec.com.edgekey.net]					
23.57.118.161	[e4593.g.akamaiedge.net]	[wildcard.skype.com.edgekey.net]					
23.57.145.48	[e5727.g.akamaiedge.net]	[fd.rad.msn.com.edgekey.net]					
23.57.154.73	[e8011.g.akamaiedge.net]	[wildcard.msads.net.edgekey.net]					
23.57.157.254	[e7768.g.akamaiedge.net]	[sancert.skypeassets.com.edgekey.net]					
23.61.187.27	[e8218.ce.akamaiedge.net]	[ocsp.ws.symantec.com.edgekey.net]					

Figura 13. Hosts acessados capturados pelo *NetworkMiner*.

4.4.3. Filtragem de imagens

Nesta outra amostra é possível visualizar as imagens remontadas pelo *NetworkMiner* a partir dos pacotes capturados no *Wireshark*.

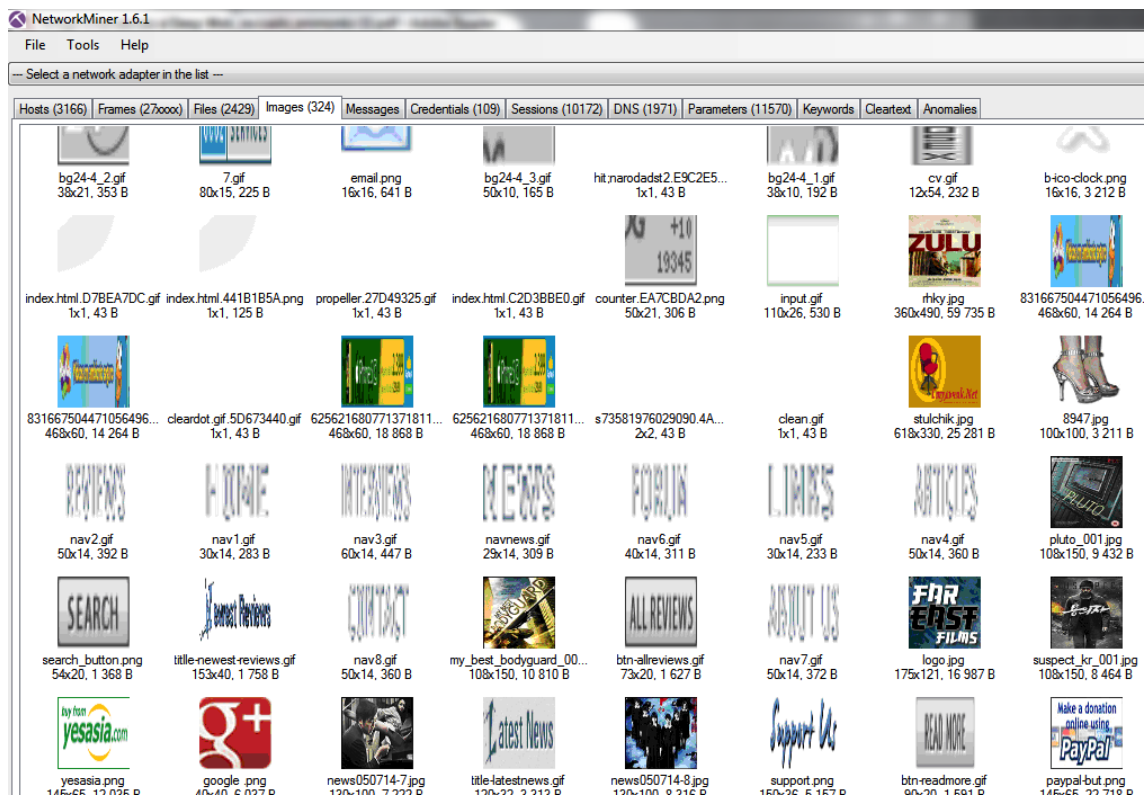


Figura 14. Imagens remontadas a partir de pacotes capturados.

4.4.4. Filtragem de arquivos

Nesta outra amostra é possível é exibido o conteúdo arquivos capturados e remontados conforme veremos nas figuras 15 e 16. O arquivo capturado foi um .xml [XML 2008] que as vezes carrega informações importantes de configurações de serviços, usuários e senhas. Pode revelar informações de vulnerabilidades de um serviço ou cliente por que seu conteúdo é escrito em texto plano e é utilizado por diversas linguagens de programação sendo o seu uso bem popular.

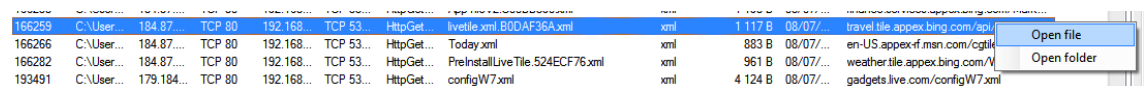


Figura 15. Lista de arquivos com a opção da visualização de conteúdo aberta.

![Screenshot of a web browser showing XML code. The browser address bar shows 'C:\Users\Sime\Downloads\NetworkMiner_1-6-1\Netv...'. The XML code is displayed in a monospaced font with syntax highlighting. It starts with '<?xml version=](http://appexblu.stb.s-msn.com/usappex/i/DC/DB49D37C7889BE7DF65D3C6DAEE1E.jpg)

```
<?xml version="1.0"?>
- <tile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <visual lang="en-us" version="2">
- <binding fallback="TileSquarePeekImageAndText04" template="TileSquare150x150PeekImageAndText04">
- <text id="1">Ghosts in Savannah, charm in Charleston</text>
- <image id="1" src="http://appexblu.stb.s-msn.com/usappex/i/DC/DB49D37C7889BE7DF65D3C6DAEE1E.jpg"/>
- </binding>
- <binding fallback="TileWideImageAndText01" template="TileWide310x150ImageAndText01">
- <text id="1">Ghosts in Savannah, charm in Charleston</text>
- <image id="1" src="http://appexblu.stb.s-msn.com/usappex/i/89/27FBE6EBB62164FAE3FBDC7E17D1E2.jpg"/>
- </binding>
- <binding template="TileSquare310x310ImageAndText01">
- <text id="1">Ghosts in Savannah, charm in Charleston</text>
- <image id="1" src="http://appexblu.stb.s-msn.com/usappex/i/DF/939FE7E799FBE2DC5585F06F5A78F9.jpg"/>
- </binding>
- <binding template="TileSquare70x70ImageAndTextOverlay01">
- <text id="1">Ghosts in Savannah, charm in Charleston</text>
- <image id="1" src="http://appexblu.stb.s-msn.com/usappex/i/DC/DB49D37C7889BE7DF65D3C6DAEE1E.jpg"/>
- </binding>
- </visual>
- </tile>
```

Figura 16. Arquivo .xml aberto expondo informações.

5. Conclusões

Usar o navegador TOR e acessar a Deep Web é fácil bastando apenas instalar o software, não exigindo nenhuma configuração extra. Já configurar um relay de saída exige também a abertura de portas no dispositivo que faz o NAT para a rede local em caso de não estar usando diretamente um ip externo.

O *Hidden Service Protocol* é o responsável por fazer todas as conexões entre os circuitos, criar pontos de encontro entre clientes e servidores, publicar na hashtable distribuída entre os relays novos serviços disponibilizados na rede TOR. Infelizmente a rede TOR sofre com alta latência gerada pelos inúmeros saltos entre diversos roteadores e relays de distâncias diferentes tornando lento seu acesso.

A *Deep Web* tem um importante serviço hoje que é promover o anonimato dos usuários quando necessário. Porém esse anonimato só é efetivo se a rede TOR for usada para acessar serviços dentro da rede TOR. A tentativa de se tornar anônimo usando o navegador TOR para acessar serviços fora da *Deep Web* é um ledô engano, já que como foi visto nos testes, se o administrador do *relay* de saída capturar os pacotes e minerálos, será possível capturar informações de imagens, usuários, senhas, endereço de hosts e arquivos. É necessário usar além da rede TOR uma conexão criptografada para esse tipo de acesso, caso contrário não há segurança.

6. Referências

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

Firefox (2014). Mozilla Firefox Documentation. Disponível em: <https://developer.mozilla.org/en-US/Firefox>. Acesso em: 10/07/2014.

Hashtable (2011). Conceito Hashtable. Disponível em: <http://www.ime.usp.br/pf/mac0122-2002/aulas/hashing.html>. Acesso em: 10/07/2014.

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

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

Network Miner(2014). Disponível em: <http://sourceforge.net/p/networkminer/wiki/Home/>. Acesso em: 10/07/2014.

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

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

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

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

XML (2008). Disponível em: <http://www.w3.org/TR/REC-xml/>. Acesso em 10/07/2014.