FACULDADE DE TECNOLOGIA DE AMERICANA – MINISTRO RALPH BIASI

Curso Superior de Tecnologia em Segurança da Informação

Thiago Bortoloto

Tor além da *Deep Web*:

Análise e comparação do *Tor* e outras ferramentas para o anonimato on-line

Americana, SP

2020

FACULDADE DE TECNOLOGIA DE AMERICANA – MINISTRO RALPH BIASI

Curso Superior de Tecnologia em Segurança da Informação

Thiago Bortoloto

Tor além da *Deep Web*:

Análise e comparação do *Tor* e outras ferramentas para o anonimato on-line

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em 2020, sob a orientação do Prof. Me. Wagner Siqueira Cavalcante

Área de concentração: Criptografia.

Americana, SP.

2020

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

B748t BORTOLOTO, Thiago

Tor além da Deep Web: análise e comparação do Tor e outras ferramentas para o anonimato on-line. / Thiago Bortoloto. – Americana, 2020.

85f.

Monografia (Curso Superior de Tecnologia em Segurança da Informação) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Ms. Wagner Siqueira Cavalcante

1 Segurança em sistemas de informação 2. VPN I. CAVALCANTE, Wagner Siqueira II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.518.5

Thiago Bortoloto

Tor além da *Deep Web*:

Análise e comparação do *Tor* e outras ferramentas para o anonimato on-line

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Segurança da Informação pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.

Área de concentração: Criptografia.

Americana, junho de 2020.

Banca Examinadora:

_____
Prof. Me. Wagner Siqueira Cavalcante
Mestre
Faculdade de Tecnologia de Americana

_____
Prof. Me. Rogério Nunes de Freitas
Mestre
Faculdade de Tecnologia de Americana

_____
Prof. Me. Maxwel Vitorino Da Silva

Mestre

Faculdade de Tecnologia de Americana

AGRADECIMENTOS

Agradeço a todas as pessoas que me ajudaram na realização desse trabalho.

DEDICATÓRIA

Aos meus familiares por todo o auxílio que me deram durante essa jornada.

RESUMO

Durante um longo tempo houve um misticismo em relação ao funcionamento do programa *Tor*, em especial como o *Tor Browser Bundle* , com sua ligação aos termos genéricos, *Deep Web* e similares, causando uma dificuldade de entendimento dos verdadeiros objetivos desse projeto e de outros *softwares* concorrentes que podem serem usados como substitutos do mesmo de acordo com as circunstâncias, sendo os principais exemplos: *proxies*, *VPNs* e *I2P*. Para a compreensão do funcionamento sobre o aspecto de *Darknet* dessa ferramenta é necessário entender seu projeto inicial e sua diferença com a alternativa mais comum a ela, *I2P*, para entender que não existe o conceito popular chamado de camadas da *Deep Web*, mas a utilização de ferramentas que utilizam *pseudo-top-level domain* com um recurso diferencial, o qual cada organização coloca um nível de prioridade diferente nesse quesito. Além de desmistificar o que é a *Deep Web*, é necessário desmistificar o funcionamento e as capacidades do *Tor Browser Bundle* comparando com ferramentas para navegação na *Web* que também ocultam o *IP* verdadeiro da conexão local, visto que também apresentaram pontos positivos em relação em diversos aspectos que foram não foram priorizados pelo *Tor*, como a característica da esteganografia da conexão. Paralelamente como os testes dos serviços de anonimato, também foram analisados os navegadores em uma avaliação não planejada, tendo como resultado uma superioridade do *Google Chrome* sobre *Firefox* na questão dos vazamentos de *WebRTC*. O resultado dos testes junto um estudo da documentação do *Tor*, é visível que o mesmo é uma opção mais segura que *VPNs* ou *proxies* em ambientes que não tente filtrar o mesmo e não use UTP enquanto o *I2P* se encontra muito limitado por tentar prover todos recursos básicos da *Internet*, além ser menos amigável para iniciantes do que o *Tor*.

Palavras Chave: *Tor*; *Proxy*; *VPN*.

ABSTRACT

For a long time had exist a mysticism about how the Tor software works in especially with the Tor Browser Bundle and it's connection with generic terms as Deep Web and similarities, making difficult to understand of true objectives of this project and other competitors software that can be use as substitutes from Tor in according to the circumstances, being the main examples: proxies, VPNs and I2P. For a better compression of how works under aspects of darknet feature of this program is necessary understand it's original project and his difference with the most common alternative to Tor, I2P, to understand don't exist levels of Deep Web as most of people think but the use of software that use pseudo-top-level domain as a differential feature, which each project put a different level of priority in this point. Beyond just demystify Deep Web, is necessary demystify how it works what are their capabilities of Tor Browser Bundle in comparison with others tools to hidden the true IP of the connection while browsing since they also had presented positives points in many aspects that don't had receive priority by the Tor Project as connection steganography a the feature. In parallel with the anonymity tests, the browsers were also analyzed in an unplanned way by the same tests, having as results a superiority of Google Chrome over Firefox in the aspect of WebRTC leek. The results of the tests and the study of Tor's documentation show software is a safe option than VPNs or proxies in cases than it is not filtered or don't have need to use UTP while I2P is very limited situation for trying to make all basics software on Internet and be less friendly for beginners than Tor.

Keywords: *Tor; Proxy; VPN.*

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

IP: Internet Protocol, é o protocolo que permite a comunicação dos computadores através da enumeração deles. Essa sigla foi usada para descrever do endereço de IP

TCP/IP: foi usado para se referenciar modelo de divisão das camadas de uma rede

OSI: Open System Interconnection, outro modelo de divisão de uma rede camadas

ARPANET: Advanced Research Projects Agency Network, uma das primeiras redes de computadores e predecessora da Internet. Também foi a primeira a utilizar os protocolos TCP/IP.

UCLA: Universidade da Califórnia em Los Angeles

UCSB: Universidade da Califórnia em Santa Barbara

NORSAR: Norwegian Seismic Array, um observatório sismológico na Noruega que foi um dos primeiros membros da ARPANET fora dos EUA.

MILNET: Military Network, rede militar que cresceu dentro da ARPANET, porém acabou se separando

NIPRNET: Non-classified Internet Protocol Router Network, uma rede do governo americano para troca de dados não confidenciais. Sucessora da MILNET

NSFNET: National Science Foundation Network, rede governamental americana com a qual a ARPANET foi absorvida. Em 1991, com mudança na legislação ela permitiu o uso comercial dela fazendo era virar a Internet como nós conhecemos.

W3C: World Wide Web Consortium organização que tem o foco em propagar padrões na Internet.

URI: Uniform Resource Identifier

URL: Uniform Resource Locator, o endereço na rede de um recurso informático

HTML: Hypertext Markup Language, linguagem de marcação que serve com base para criação de páginas na Internet

HTTP: Hypertext Transfer Protocol, o principal protocolo utilizado para transmissão de dados na Web.

HTTPS: Hyper Text Transfer Protocol Secure, semelhante ao HTTP, porém é criptografado

FTP: File Transfer Protocol, protocolo utilizado para transferência de arquivos.

SSH: Secure Shell, um protocolo que serve de base para programas de acesso remoto com conexão criptografada.

DNS: Domain Name System, protocol

Tor: The Second-Generation Onion Router, é a segunda geração de onion routing que utilizada para garantir privacidade e anonimato da conexão. Também permite acessar endereços de domínio .onion.

I2P: Invisible Internet Project, uma rede criada para trabalhar de forma paralela à Internet. Diferente do Tor ela tem como objetivo abrigar seus usuários dentro dela, ao invés de permitir que eles transitem por endereços de domínio comum.

VPN: Virtual Private Network, é protocolo que especifica como criar uma conexão segura entre um elemento em uma rede pública para que o mesmo tenha acesso a uma rede privada. Foi utilizando para descrever também empresas que oferecem uma conexão seguram para diversos servidores que fazem de intermediário na conexão entre o usuário e recurso solicitado na Internet.

JAP: Java Anon Proxy, também conhecido como JonDonym é programa que que trabalha de intermediário entre o usuário e a Web. Necessita um navegador especial para o uso.

ICANN: Internet Corporation for Assigned Names and Numbers, é uma organização não governamental que gerencia comercialização e funcionamento dos DNS.

SSL: Secure Sockets Layer, Protocolo de segurança para encriptar conexões HTTPS.

TLS: Transport Layer Security, um novo protocolo de segurança que desempenha as funções do SSL.

IRC: Internet *Relay* Chat, é um protocolo para comunicação on-line. Com ele é capaz de criar servidores que hospedam salas de chats e acessá-las de forma parecida com um servidor Web e um navegador.

P2P: Peer-to-peer, é uma arquitetura de rede decentralizada onde seus computadores trocariam arquivos entre si. Esse modelo é muito utilizado em redes de trocas de arquivos como a GNUnet.

F2F: Friend-to-friend, similar ao P2P, porém o usuário escolhe que serão seus contatos que intermediaram sua conexão com resto da rede.

OR: Onion Router, modo de chamar os nós que participam a rede do Tor.

OP: Onion Proxy, é o programa que gerencia o acesso e uso da rede Tor.

CircID: Circuit Identifier, número no cabeçalho que identifica o ID do circuito entre dois nós da rede.

TCP: Transmission Control Protocol, é o protocolo de rede que trabalha na camada de transporte. Ele verifica que os dados transmitidos chegaram seu destino retorna essa informação ao remente.

UDP: User Datagram Protocol, é o protocolo de rede que trabalha na camada de transporte, porém não faz a verificação do envio dos dados. Usando mais para streaming.

AES: Advanced Encryption Standard, principal algoritmo de criptografia simétrica que é o padrão de mercado atualmente.

WebRTC: Web Real-Time Communication, é um protocolo para comunicação em tempo real através de áudio e vídeo com grande foco em dispositivos moveis e navegadores.

EFF: Electronic Frontier Foundation, é uma organização não governamental que defende direitos digitais como direito compartilhamento de arquivos, criptografia, privacidade etc.

1 INTRODUÇÃO

A razão pela qual esse trabalho foi iniciado, foi por causa da mitificação que se desenvolveu em volta de ferramentas que prometem o anonimato *on-line*, como o *Tor* e o *I2P*. Os termos como *Deep Web* e *Darknet* foram usados de forma não organizada pela mídia e pouco foi explicado sobre o funcionamento destes programas, sendo o exemplo mais sensível é que o *Tor*, que foi desenvolvido com foco de burlar medidas que causem bloqueios em páginas na *Web* em endereços de domínio comuns, enquanto que o uso dos domínios *.onion* são um objetivo secundário.

Devido ao tempo dessa representação equivocada por parte da mídia, sendo criada por falta de especialistas ou para atrair atenção para as matérias, essa atitude no mínimo intensificou três grandes problemas: a questão de semântica, dúvidas de como o *Tor* e o *I2P* funcionam, e quais são suas reais capacidades.

Esses dois últimos pontos levantam o questionamento da razão de necessidade usar o *Tor* e o *I2P*,s se já existem outros programas parecidos que proporcionam uma certa proteção ao anonimato na *Internet*.

Em um cenário positivo, ambos os projetos seriam uma versão aperfeiçoada da *Internet*, oferecendo diversos complementos técnicos que não poderiam ser emulados por serviços comuns da *Internet*, além de entregarem uma maior segurança e privacidade por terem essas características desde seus projetos iniciais.

Caso a situação seja oposta, a *Internet* comum continua sendo uma opção superior em comparação a ambas as redes*,* sendo capaz de realizar as mesmas atividades como um grau de segurança similar ao dos serviços disponíveis na mesma, o que deixa essas redes em uma situação redundante.

O meio termo entre ambos os cenários seria no caso que as duas redes trabalhariam para funções específicas, com um rendimento inferior aos dos serviços atuais. Algumas das características delas poderiam ser emuladas por programas comuns, porém de forma limitada ou com rendimento inferior. O *Tor* e o *I2P* seriam ferramentas que trabalhariam de forma complementar aos serviços mais comuns e usadas na forma à qual seus criadores recomendam.

O objetivo principal deste trabalho é explicar os objetivos de ambos os projetos em seu escopo inicial e compará-los com outros recursos para ocultação do endereço de *IP* na *Internet*.

Os objetivos secundários são os de situar onde se encontrariam as redes do *Tor* e *I2P* na *Internet* de forma geral, mostrar como palavras para taxar como ambos projetos são vistos e usados em diferentes pontos de vista, mostrar de forma breve o funcionamento de um serviço oculto do *Tor*, e customizar algumas ferramentas para ver o quão próximo as qualidades delas chegam dos resultados da dupla.

O trabalho começa por uma introdução, que corresponde à parte teórica, explicando brevemente sobre quais pilares a *Internet* e a *Web* foram construídas. Ainda no mesmo capítulo são apresentadas as diversidades de termos como *Deep Web*, *Darknet* e outros, explicando seus significados para diferentes pontos de vistas. Na sequência do mesmo capítulo, explica-se o que são o *Tor* e o *I2P*, como foram idealizados e para que foram planejadas, segundo seus artigos de publicação originais e como eles funcionam atualmente.

O segundo capítulo é dividido em um breve resumo das ferramentas secundárias, a primeira bateria de teste, teste com um *hidden service* e os testes finais. Na apresentação de ferramentas secundárias, explica-se o que são *proxies*, *VPNs* e *JAP* e seus diferentes subtipos.

O terceiro capítulo tem apresentação detalhada dos recursos utilizados para a análise, seguida pela exposição das tabelas de resultados dos testes *on-line* de terceiros para a busca de problemas de segurança e capacidade de identificar a utilização do serviço das ferramentas principais e secundárias. Na sequência tem-se uma breve análise da transmissão de dados através de um *hidden service* da rede *Tor* através da utilização do *Wireshark*.

O trabalho finaliza com testes das ferramentas secundárias, configuradas para apresentar uma maior privacidade.

2 HISTÓRICO DAS *DARKNET*S

Quando se pergunta sobre o que é *Deep Web*, é necessário primeiro explicar o que é *Web* e a sua diferença com *Internet* para posteriormente entender os significados do "*Deep"*. Segundo o W3C, a *Internet* surgiu sob a estrutura criada pela ARPANET, uma rede de redes de computadores. Já a *Web* é um dos serviços oferecidos pela *Internet* através de um conjunto de tecnologias no qual se destaca o protocolo *HTTP*, *URL* e a linguagem de marcação *HTML (*W3C, [s.d.]*).*

A mesma linha de pensamento também é reafirmada na Cartilha de Acessibilidade na *Web* que é distribuída on-line pela W3C Brasil. Nela é explicado que *Web* é um dos vários serviços que a *Internet* pode oferecer e menciona sobre sua evolução de um conjunto de páginas estáticas para páginas dinâmicas. Ainda é explicado que a *Web* é formada por uma gigantesca quantidade de documentos dos mais variados formatos que são indexados nas páginas da *Web (*W3C BRASIL, [s.d.]*)*.

Esse conceito pode ser sumarizado dizendo que a *Web* é uma parte da *Internet* onde tem o conteúdo multimídia *(*MORAIS; LIMA ;FRANCO, 2012, p.58-59*).*

Para uma melhor compressão do estado presente da *Internet* é necessário entender a sua predecessora, a *ARPANET*. Uma das suas principais características, a descentralização, gerou o mito que mesma foi projetada para ser capaz resistir a um ataque nuclear. No livro Impérios da Comunicação *(*WU, 2012, p211-212) explica-se que parte dessa narrativa se deve a um homem, Paul Baran, que queria uma maneira de um país continental como os EUA poder se comunicar após um cataclismo dessa escala e se redirecionasse para a recuperação econômica.

Porém, o sistema telefônico monopolista de longa distância da Bell era extremamente vulnerável em caso de um ataque soviético (na época da guerra fria), devido a da sua falta de redundância e extrema centralização que já vinha no seu *design* pela idealização de eficiência e economia de recursos. Era uma questão de conflito conceitual entre um sistema centralizado e um descentralizado *(*WU, 2012, p211-2*)*.

Os dois primeiros membros da *ARPANET* foram a UCLA e o instituto de pesquisa de Stanford, os quais fizeram a primeira conexão, e posteriormente terceiro

membro a ser adicionado nessa rede foi um computador na UCSB seguido por outro na universidade de Utah *(*Stewart, 2000). A primeira instituição da rede que não era americana é disputada pela Norwegian Seismic Array através de uma conexão por link de satélite *(*NORSAR, [s.d.]) e primeira conexão internacional cabeada da *ARPANET* foi com Londres *(*Stewart, 2000*).*

Um elemento importante a ser considerado para um sistema descentralizado como a *ARPANET* é a necessidade de todos os equipamentos de se comunicarem sob um mesmo protocolo de rede. Esse protocolo seria o *IP*, *Internet* Protocol, surgiu para facilitar a comunicação em uma rede proprietária, sobre a qual *ARPANET* não tinha controle. Através do conceito de encapsulamento era possível enviar em qualquer estrutura física e posteriormente isso mais tarde levaria às arquiteturas em camadas de rede *TCP/IP* e *OSI (*WU, 2012, p238-240*)*.

Em 1983, a ARPANET foi dívida em duas redes, uma com proposito militar, que se chamaria MILNET, e outra, com propósito civil, que continuaria sendo chamada de ARPANET. Esse processo de divisão teria quatro etapas.

Em 4 de abril, criou-se a MILNET e, nós ligados ao Departamento de Defesa norte americano seriam designados para ela, porém continuariam a dividir o mesmo *backbone*. Em 4 de outubro, as duas redes se tornaram separadas, cujos serviços serviriam como ponte de comunicação entre ambas para pessoal autorizado.

Em 1º de dezembro foi configurado um software de controle de acesso na MILNET. No primeiro trimestre de 1984 foi finalizado o processo de separação física de ambas as redes. Existiriam 3 comunidades especializadas dentro de cada rede. A comunidade da ARPANET passou a focar em experimentação e testes, sendo que estabilidade das comunicações seria um objetivo secundário para ela.

Já a MILNET contou com duas comunidades, uma aberta, que se comunicaria com a ARPANET e teria um foco mais operacional que experimental, e outra, fechada, formada por *hosts* militares que não teriam necessidade de se comunicar como a ARPANET (NIC, [s.d.]).

Nos anos 90 a *MILNET* foi transformada em uma rede mais moderna, a *NIPRNET (*GLOBO, 2009), que atualmente é a rede de menor importância de comunicação interna norte americana *(*WEINBERGER ,2010,p.2*) .* A *ARPANET*

seguiria o mesmo caminho sendo desativada nos anos 1990 e seus membros relacionados a pesquisa migrados para a *NSFNET*. Em 1995 o *NSFNET* foi descomissionado e as últimas regras que atrapalhavam a *Internet* comercial foram removidas *(*STEWART ,2000).

O próximo ponto ser analisado, o *URL* (*Uniform Resource Locator*), é um esquema de endereçamento. Sua sintase é a sigla do protocolo seguido por um endereço dos servidores, a porta do serviço e o caminho do arquivo que ser alcançado. No caso da *Web*, o protocolo a ser utilizado será o *HTTP*, outro elemento básico dela (BERNERS-LEE, 1994). Pode-se ter *URLs* dentro da Internet ao mesmo tempo. Ele está fora da *Web*, como é o caso dos serviços de e-mail.

Um recurso que potencializou o funcionamento do *URL* foi *Domain Name System*, o *DNS*, o serviço que faz a conversão do endereço de domínio para o *IP* do servidor.

Começou a ser desenvolvido como uma substituição do antigo *Host Table,* que tinha as funções de eliminar a necessidade do usuário saber o endereço numérico de onde ele iria se conectar, promover estabilidade em caso de mudança de endereço, e a capacidade de vários endereços estarem associado um único *host*. O *DNS* teria de preservar a capacidade de utilização da *Host Table* ao mesmo tempo deveria ter um *design* que iria promover uma estrutura hierárquica sólida, robusta e distribuída *(*KLENSIN, 2003, p.1*).*

Cade Metz *(*2012) explica em sua matéria no Internet Hall of Fame que antes da implementação do serviço de *DNS*, caso alguém quisesse adicionar uma máquina na *ARPANET*, essa pessoa teria que entrar em contato com o *Stanford Research Institute Network Information Center*. Isso limitava a capacidade de trabalho na rede de acordo com o horário de funcionamento do instituto.

Outra vantagem do *DNS* que o sistema iria introduzir seria a capacidade de distribuir o serviço de relacionar o endereço virtual com o número de *IP* entre diversos servidores. Isso permitiria um maior dinamismo das redes já que permitiria a atualização do nome do *host* e seu *IP* associado no *DNS* local sendo que esse traduziria para as requisições internas e externas.

Atualmente segundo a equipe do *site inetdaemon (*2018), a hierarquia de *DNS* pode ser dividida nos seguintes níveis: em *Root*, *Top Level Domains*, *Second Levels Domains*, *Sub-Domains* e o *Host Name* (Raiz, Domínios de nível superior, Domínios de nível secundário, Subdomínios e Nome de Hospedeiro).

Os servidores *Root DNS* são aqueles que estão acima na hierarquia dos servidores de domínios como .org ou .br. Eles são o primeiro passo em uma pesquisa de um endereço de domínio *(*KARRENBERG, 2004, p.4-5*).*

O W3C Brasil define o *HTML* como uma linguagem de marcação que serve para escreverá estruturas das páginas como *hiperlinks*, títulos, tabelas, formulários e outros enquanto *(*FIELDING ET AL, 1994, p.1-2) o protocolo *HTTP* é utilizado para transferir esse hipertexto.

O *HTTP* é um protocolo na camada de aplicação com o principal objetivo de transportar dados no formato de hipertexto e hipermídia começando a ser usado na *Web* por volta dos anos 90. Antes da versão 1.0, o protocolo transmitia os dados em um formato *RAW*, mas após a primeira versão estável começou transmitir em formato *MIME* contendo metainformação e posteriormente o protocolo foi refinado em sua versão 1.1 *(*FIELDING ET AL,1999,p. 7*) .* Devido a utilização para aplicações com dados sensíveis foi necessário utilizar protocolos para criptografar os dados que era transmitidos por *HTTP (*RESCORLA, 2000*).*

2.1 *DARKNET*, *DEEP WEB* E *DARK WEB*

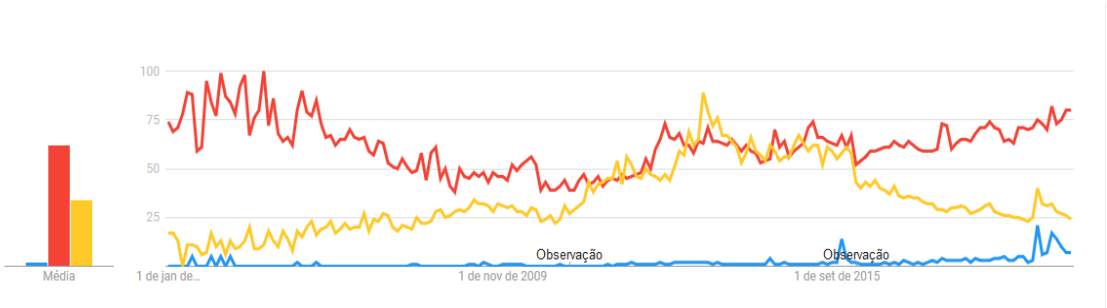
Em *The Darknet and the Future of Content Distribution* argumenta que o conceito de *Darknet* como uma rede tendo como suas três principais características que seriam as capacidades para seus de disponibilizar arquivos entre eles, permitir a cópia dos arquivos da rede e deixarem seus integrantes conectados em canais de banda larga.

Nesse artigo existe uma forte ligação entre o conceito que eles definem de *Darknet* com o ideia de pirataria sendo que utilizaram o exemplo de *Sneakernet*, a troca de material pirata pessoalmente, servidores centralizados de *FTP* para compartilhamento de arquivos de *mp3*, o *Naspter* e o *Gnutella* como exemplos de *Darknet (*PETER BIDDLE ET AT., 2002, p.1*)*.

Essa opinião também é articulada de forma similar no *artigo The Darknet: A Digital Copyright Revolution* que definiu o termo *Darknet* como as redes de compartilhamento de arquivos de forma clandestinas que ofereciam segurança, privacidade e formas anônimas de distribuir arquivos *(*WOOD, 2006, p.1*).*

Outro termo tão usado quando *Darknet* é *Deep Web*. Para tentar entender o que seria a *Deep Web* na cabeça de uma pessoa comum seria ilustrativo procurar o momento que ela teve o primeiro contato como termo. Para tentar aproximar disso, um bom momento é verificar o pico de popularidade do termo no *Google Trends*. Esse evento aconteceu em 20 de março de 2013 como é demostrado na Figura 1.

Figura 1 – Comparação entre os termos anonimato, *Tor* e *Dark Web*.



Fonte: *Google Trends* (2019)

Como pode-se ver, o termo “anonimato” em vermelho foi mais popular do que o termo “Tor” durante maior parte do tempo, sendo superado por curtos períodos entre 2011 e 2015. Já o termo, “Dark Web” começou ter alguma significância a partir de 2015. Outro ponto é que termo “anonimato” pode servir, tanto para ambientes virtuais, como para pessoas que saíram da fama (o que pode indicar a popularidade detectada).

Uma apresentação parcialmente formal que tenha atingido um certo grau de popularidade na época, foi uma matéria do programa de televisão, Olhar Digital, sobre o assunto. Esse trecho do material foi recortado e disponibilizado por terceiro, sendo que o vídeo do programa original foi bem menos popular. No vídeo são apresentadas as seguintes informações:

- A *Deep Web* seria 90% de toda *Web*
- A comparação do Iceberg por causa do tamanho onde só pequena parte está visível

- O tamanho da *Deep Web* (A web na superfície é 10% e *Deep Web* é 90%)
- A questão do Iceberg
- Na *Deep Web*, nada é indexado ou rastreável
- Todo o tráfego de dados é criptografado, o que significa privacidade e anonimato.
- Na Web comum a rastreabilidade é fácil e privacidade é ilusão
- As autoridades sabem que ela existe, mas não existe nada que elas possam fazer.
- Para acessar a *Deep Web* é preciso navegador especial.
- “Vírus surgem lá” e “na dúvida, não clicar”.
- Nenhum site consegue coletar seus dados través do Tor.

Já na academia, existem visões diferentes para interpretar esse conceito, começando pela questão semântica. Segundo Monteiro e Fidencio (2013,p. 1-2), a Internet é dividida em uma superfície indexada e uma parte “imersa”, bem maior, não indexada, descrevendo-as como:

> “A informação na *Web* pode ser categorizada, para fins de indexação, em suas diretrizes: a parte visível, ou seja, páginas que podem ser somadas ao banco de dados dos buscadores, e a parte invisível, cujo conteúdo, por razões expostas, não pode ser indexado pelos buscadores tradicionais *(*MONTEIRO; FIDENCIO, 2013 ,p. 10*) .*”

Essa caracterização também é reafirmada por diversas fontes como um trabalho de pesquisa de Ciancaglini et al *(*2015) e um artigo de Singletary *(*2015) . Com essas afirmações pode-se excluir a ideia de que o conteúdo de um serviço da *Internet* seria a característica suficiente para definir se ele faz parte da *Deep Web*.

Em seu artigo, Monteiro e Vinicius *(*2013) descrevem que as primeiras tentativas de indexação na *Internet* eram feitas através de um processo manual que resultaria nos Diretórios (*Web Directory*), porém devido à rápida expansão de conteúdo foram desenvolvidos métodos automatizados de indexação que resultariam nos sistemas

dos primeiros motores de buscas, como o *Altavista* ou *Northern Light*, que são os antecessores indiretos do *Google*.

Para o pleno funcionamento desses motores de busca é necessário um tipo de programa que faça diversas varreduras na *Internet* à procura de novos elementos que podem ser indexados. Esses softwares são rotulados de robôs, também conhecidos como *Crawlers* ou *Spiders*, e nas palavras de Monteiro e Fidencio *(*2013) “sua função básica seja pesquisar, relacionar, adentrar diretórios e subdiretórios na *Web* e somá-los aos índices dos buscadores para os quais operam”.

Para auxiliar em uma indexação mais vantajosa para si, os *sites* normalmente têm um arquivo com nome de “*robots.tx*t”, no qual possui instruções de como os *bots* devem executar a indexação e de qual conteúdo que não deve ser indexado. Também é possível solicitar a não indexação por uma tag HTML (“<META NAME=”ROBOTS” CONTENT=”NOINDEX,NOFOLLOW”>)”.

Segundo Monteiro e Fidencio *(*2013), desse modo é possível um *site* solicitar para os indexadores ignorarem um conteúdo especifico, uma página ou mesmo o *site* todo, o que deveria teoricamente faz com que o mesmo não seja exibido nos resultados dos motores de busca, podendo ser rotulados como uma parte da *Deep Web* através dessas características.

Todavia, a não indexação pode ser originada por diversos métodos e motivos além de instruções de não permitir o acesso pelos robôs indexadores. Utilizando razões da falta de indexação os autores dividem a *Internet* Invisível em cinco grandes grupos como é mostrado na Figura 2: *Internet* Opaca, *Internet* Privada, *Internet* Proprietária, *Internet* verdadeiramente invisível e a *Dark Web*.

A *Internet* Opaca é a fronteira entre *Internet* Visível e a Invisível. A maior parte das causas da opacidade para indexação são antigas limitações técnicas como o alcance e a frequência dos rastreadores e pela falta de links e conexões.

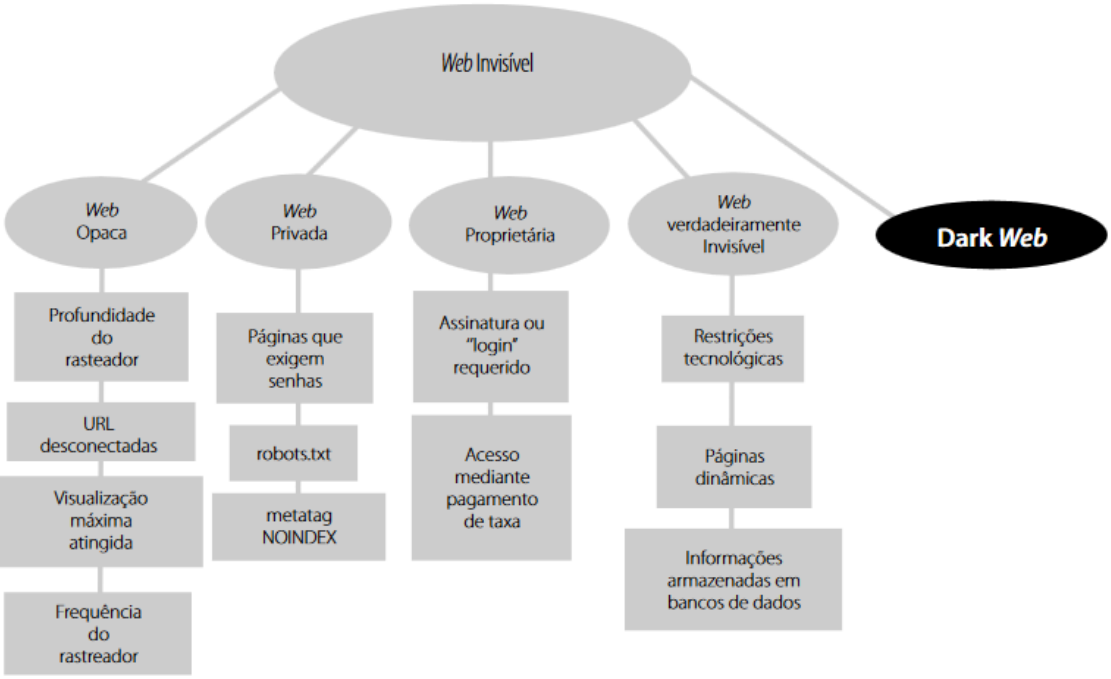
Já a *Internet* Privada são os *sites* e conteúdo que são protegidos por requisições de login e configurações para que certos conteúdos não sejam indexados.

No caso da *Web* Proprietária existe necessidade da assinatura do conteúdo que pode ser gratuito ou com uma forma de pagamento para ter o acesso ao mesmo.

A “*Web* Verdadeiramente Invisível” é causada por incapacidades técnicas, tendo como exemplos os formatos de certos tipos de arquivos, políticas de exclusão e incapacidade de indexação de páginas dinâmicas.

Por último é explicada *Dark Web* ou *Darknet* que é definida como: “Rede global de usuários e computadores que operam à margem da visibilidade e das agências fiscalizadoras, com conteúdo intencionalmente escondidos e protocolos de comunicação inacessíveis para um sistema sem configuração correta,” e cita a rede *Freenet* como um exemplo, explicando que uma parte do disco rígido de um computador conectado com a rede é reservada para armazenar arquivos criptografados de outros membros da rede.

Figura 2 – Ramificações da *Web* Invisível



Fonte: Monteiro e Fidencio (2013)

Então através da Figura 2 percebe que o termo *Deep Web* é mais abrangente do que redes ocultados como o *Tor.*

Em um artigo, Ciancaglini et al (2015) também defende que a *Dark Web* seja um tipo especial de *Deep Web*, comparando a *Deep Web* como uma mina e a *Dark Web* como uma parte especial da mina, a qual precisa de equipamentos especiais para se adentrar, (no caso, programas como o *Tor*, *I2P* e *Freenet*), que são chamadas também de *Darknets*.

O trabalho de pesquisa da *Trend Micro* cita outra característica marcante das *Darknet*s, que é o uso de domínios que não possuem ligações com o *ICANN*. Taxadas como redes de acesso limitado, pode-se dizer que a mesma necessita de servidores de *DNS* específico para resolver as requisições de domínio necessárias. Como exemplo de alternativa aos modelos de *DNS* do *ICANN* há .BIT que funciona de uma forma mais descentralizada.

A mesma pesquisa da Trend Micro, aponta aspectos que tornam a *Dark Web* complicada para atuação das forças policiais:

- Uso abrangente de criptografia
- A dificuldade de atribuição
- Variação dos endereços na rede

Singletary (2015) aponta que a capacidade de indexação do *Google* é estimada entre 4% até 40% da *Web*. Também argumenta que sites que se encontram na Internet Profunda podem contar com a falta de “linkagem” com outras páginas, em que esta não aponta para ninguém e ninguém aponta para esta. O autor também confirma a afirmação da utilização de outros tipos de protocolos que não o *HTTP* ou sua versão criptografada, o *HTTPS* (*HTTP* Seguro).

Na visão do *Singletary* (2015) afirma que redes como o IRC e dos programas P2P também são exemplos de *Darknets*, dando uma abrangência maior ao termo. Entretanto, no caso do IRC, explica-se que a maioria dos servidores desta rede não usam criptografia para proteger as transmissões dos dados com seus clientes, apesar dos diversos métodos de segurança contra interceptação de dados como o *SSL/TLS* ou *Off-Record* para garantir um aumento da privacidade.

Em relação a cronologia das *Darknet*s, entre as mais famosas encontra-se a *Freenet* que começou a ser desenvolvida em julho de 1999 *(THE FREENET PROJECT*), porém só iria se tornar uma *Darknet* verdadeira em 8 de maio de 2008 através da atualização 0.7 *(*IHLENFELD, 2008*).* Depois dela, como “grandes” projetos, houve a *GNUnet em 2001 (GNUnet* e.V.) e o *Tor* em 2002 *(*Roger Dingledine ,2002) e o *I2P* em 2003 *(I2P* TEAM*).*

Apesar dos quatros sistemas exemplificados por serem *Darknet*s, eles trabalham de forma diferentes. *GNUnet* e a *Freenet* funcionam através de conexões do tipo Friend-to-Friend.

Segundo Dan Bricklin *(*2000) o conceito de *Friend-to-Friend* é uma adaptação do *Peer-to-peer* que ao invés do usuário se conectar com prováveis desconhecidos, a conexão ocorre com grupos de “amigos” já conhecidos. Entretanto, esse modelo tem como fraqueza de não se tornar evidente quem são estes “amigos” conectados.

Porém, apesar das diferenças de funcionamento, essas quatro redes têm um ponto em comum de grande importância: elas são de baixa latência. Isso significa que elas trabalham com tempo de resposta relativamente rápido.

No artigo *Towards Efficient Traffic-analysis Resistant Anonymity Networks* redes de alta latência tem uma velocidade consideravelmente menor e tem um atraso que somente permitirá o uso de programas de comunicação similares ao e-mail. Em compensação encontra uma resistência para relação a ataques de análise de tráfego. Já as redes de baixa latência permitem a utilização de serviços de rede como o navegador *Web* ao custo de uma redução sensível na garantia de anonimato e fragilidade contra um adversário global *(*HOPPER; VASSERMAN; CHAN-TIN, 2010, p.4*).*

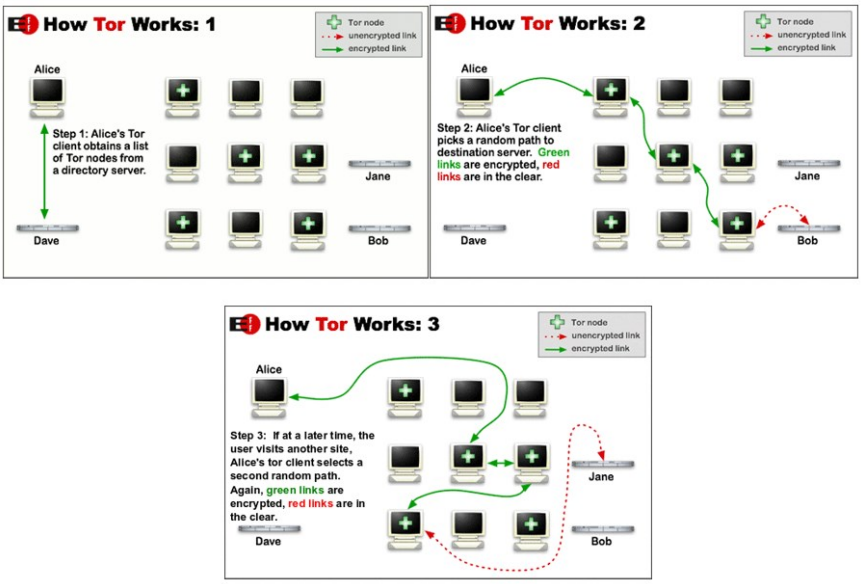
2.1.1 *TOR* SECOND-*GENERATION ONION ROUTER*

Atualmente, quando alguém usa o termo *Deep Web*, normalmente estará falando sobre o *Tor (ANTICOUNTERFEITING COMMITTEE*, 2015, p. 2*).* Nos últimos anos o *Tor* ganhou fama em relação seu uso para navegação em seu *pseudo-top-level domain*, o *.onion*, mesmo esse sendo um objetivo secundário.

Existe dois tipos de navegação utilizando o *Tor*: a que acessa endereços de domínio do *ICANN* e acessa os *pseudo-top-level domain*.

Na página do projeto tem uma explicação introdutória sobre o funcionamento do *Tor*, detalhando o funcionamento e a navegação em endereços de domínios tradicionais através da conexão com o *Tor* conforme a Figura 3. Primeiro o seu cliente consegue alguns endereços de nós em servidores de diretórios.

Figura 3 – *Tor* criando um circuito para se conectar em domínio comum

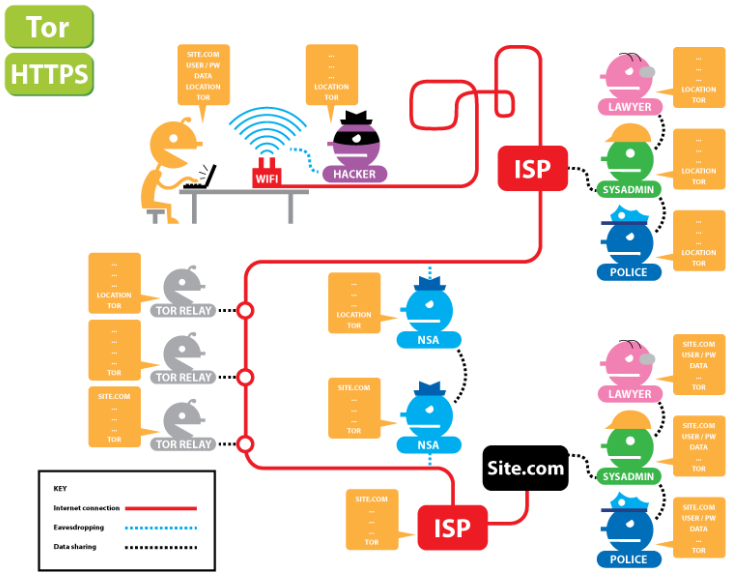


Fonte: EFF e *Tor* Project

Após isso ele escolherá um caminho randômico que arranjará a conexão entre seu cliente e o servidor o que está sendo requisitado. Todo caminho em criptografado nativamente, com exceção da conexão entre o nó de saída e o servidor do recurso requisitado, que normalmente é criptografado por *HTTPS*. Após um tempo, o circuito é alterado.

A importância disso é detalhada visualmente na Figura 4 que é uma captura de tela de uma aplicação on-line da *EFF* que exibe a situação de alguém que estiver usando o *Tor* e acessando um *site* que usa *HTTPS*.

Figura 4 – Combinação do *Tor* com *HTTPS*

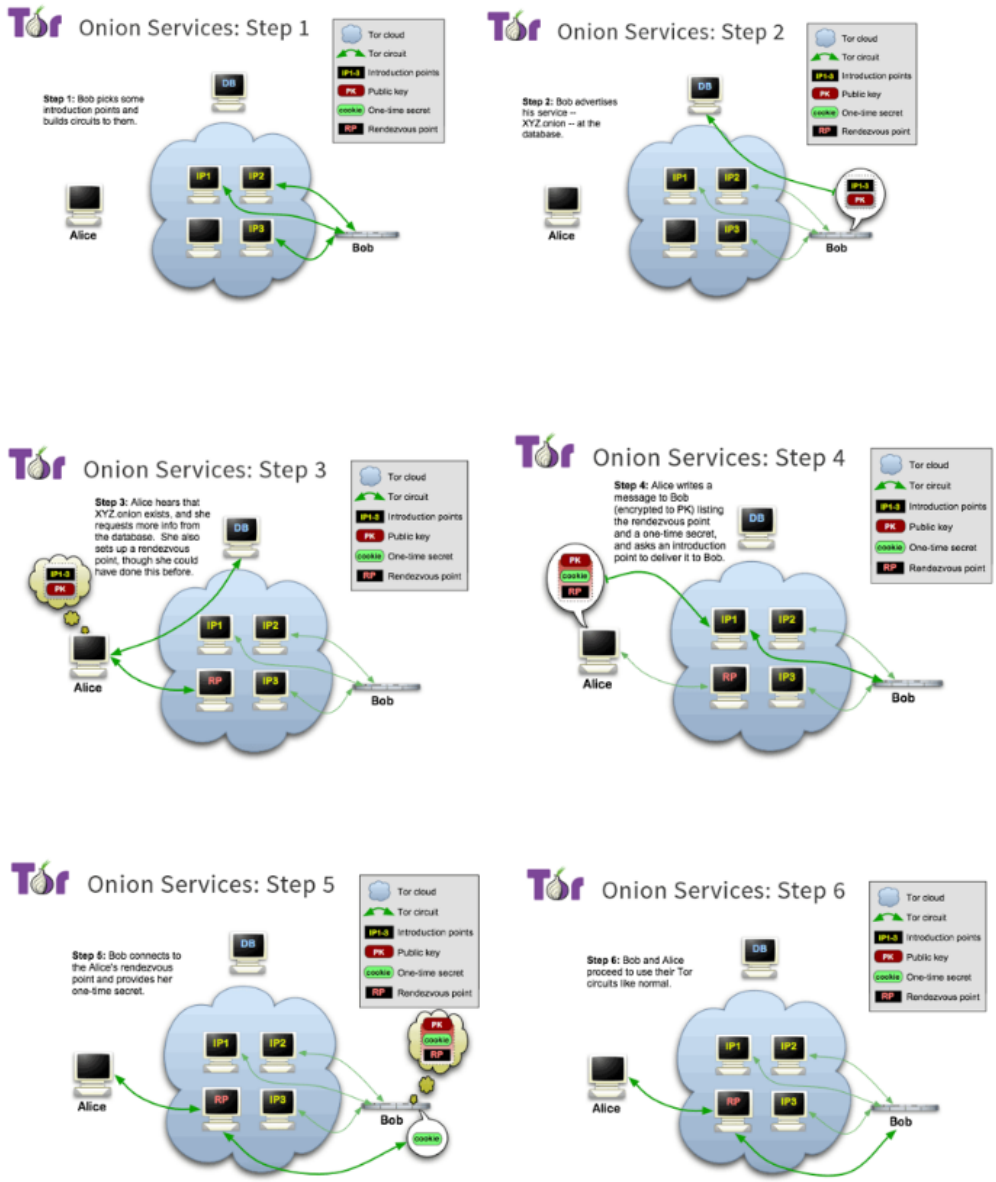


Fonte: EFF

Nela é possível observar que em caso do usuário estiver tendo suas conexões de rede analisadas por terceiros, eles só poderão saber que ele se conecta com o *Tor* e a localidade da conexão, mas não os *sites* que está acessando. Já se algum serviço estiver observando as conexões que chegam até ele, ele poderá saber diversos dados que por ser o provedor, mas não conseguira saber a localidade de uma conexão que estivesse usando o *Tor*.

Na Figura 5, apresenta a página que detalha o protocolo *Tor* no *site* do *software* explica a diferença da utilização de uma conexão com servidores que utilizam um *pseudo-top-level domain .onion*.

Figura 5 – *Tor* criando um circuito para se conectar em domínio .onion



Fonte: *Tor* Project.

Esse processo não começa com o cliente requisitando uma conexão para o servidor de seu interesse, mas como servidor escolhendo alguns nós da rede *Tor* para serem seus *introduction points*, entregando para esses nós a sua chave pública.

Após isso ele faz o *upload* da chave pública dele e a lista de *introduction points* para uma *distributed hash table*. Em algum momento o cliente vai descobrir a existência do endereço *.onion* e iria tentar acessá-lo. Com isso ele primeiramente vai acessar o *distributed hash table* e estabelecera um *rendezvous point*.

Posteriormente o cliente vai enviar uma mensagem para o servidor pedido para criar uma conexão com o *rendezvous point*. Com isso eles tem uma conexão indireta entre eles. Também é necessário notar que eles usam circuitos da rede *Tor* e não uma conexão direta em todo processo.

A sigla *Tor* serve para se referenciar a segunda geração de *Onion Routing* que foi para comunidade cientifica pelo do artigo acadêmico tendo objetivo de resolver as limitações que o *design* da primeira geração tinha através da adição dos seguintes recursos *(*DINGLEDINE; MATHEWSON; SYVERSON, 2003, p.1*).*

- *Perfect forward secrecy* (Sigilo Encaminhado Perfeito)
- Separação “*protocol cleaning*”(limpeza de protocolo) do anonimato
- Sem *mixing*, *padding*, ou *traffic shaping*
- Diversas conexões TCP devem dividir o mesmo circuito
- Topologia *Leaky-pipe circuit*
- Controle de congestionamento
- Servidores de diretório
- Variedade de política para nós de saída
- Checagem de integridade *end-to-end* (Ponta-a-ponta)
- *Rendezvous points* (Pontos de encontro) e *hidden services* (Serviços Ocultos)

Os autores explicam que esse novo *design* tentaria se equilibrar entre o anonimato, usabilidade e eficiência. Seus objetivos são descritos como facilidade de implantação, usabilidade, flexibilidade e ter um *design* simples.

A facilidade de implantação (*deployability)*, visa criar uma instância do Tor que não causasse um fardo financeiro sobre os voluntários e que não os colocariam em risco de implicações legais, que não fosse extremamente complexo de implementar. *(*DINGLEDINE; MATHEWSON; SYVERSON, 2003, p.3)

A característica de usabilidade era para facilitar a entrada de novos usuários o que causaria o aumento do anonimato devido a expansão dos números da rede. Para atingir isso seria necessário ter o menor número de decisões de configurações possíveis e não exigir modificações em programas para trabalhar junto com o *Tor* e também funcionar na maior quantidade de sistemas operacionais possíveis. *(*DINGLEDINE; MATHEWSON; SYVERSON ,2003, p.3-4)

A flexibilidade é definida como característica a ser utilizada para futuras pesquisas de forma que evitasse “reinventar” o *Tor* no futuro (DINGLEDINE; MATHEWSON; SYVERSON, 2003, p4).

O *design* simples significa manter o *Tor* com máximo de simplicidade possível, evitando adicionar recursos que possam atrapalhar sua estabilidade (DINGLEDINE; MATHEWSON; SYVERSON, 2003, 4).

O artigo também cita quatro pontos que eles não tentariam transformar em objetivos futuros do projeto. O primeiro é que eles não tentariam operar de forma ponta-a-ponto (P2P), sendo que isso significou que foi criado com um certo nível de centralização. Um outro ponto é que se admite uma fragilidade da rede contra os ataques de ponta-a-ponta (*end-to-end*), apesar da pesquisa de como lidar com essa ameaça.

O Tor também não seria capaz de fazer protocol normalization (normalização de protocolo), sendo necessário usar um recurso externo, o Privoxy, para ocultar as diferenças entre os clientes que utilizariam o Tor para se conectar. Também é explicado que não teria qualquer tentativa de fazer estereografia em relação a uma conexão do Tor (DINGLEDINE; MATHEWSON; SYVERSON, 2003,4).

Na época em que o artigo foi produzido, o modelo de um adversário global passivo era mais usado como exemplo de ameaça em mente quando era desenvolvido um software com objetivos similares ao Tor. Esse modelo não foi

considerado pela equipe do Tor Project como uma ameaça realista e foi visto como um cenário de uma rede de baixa latência que não conseguira se proteger.

Por isso o Tor foi planejado, visando enfrentar um adversário mais limitado sendo que nessa nova situação, o atacante teria como objetivo tentar observar ambos os lados da comunicação e o Tor planejaria evitar ataques de análise de tráfego do que ataques de confirmação de tráfego (DINGLEDINE; MATHEWSON; SYVERSON, 2003,4).

É descrito no artigo que o adversário sob o qual o *Tor* foi planejado, poderia utilizar como táticas de ataques tentar se passar por uma das pontas da comunicação, observar passivamente as entradas e saída de uma conexão, tentar impedir o tráfego em nós confiáveis e tentar subverter os servidores de diretório para dar aos clientes da rede uma visão diferente de seu estado atual(DINGLEDINE; MATHEWSON; SYVERSON, 2003, p.4).

O *Tor* é uma rede sobreposta que utiliza de nós da rede para executar serviços chamados *Onion Router*. Esses programas funcionariam sem precisar de permissão especial, possibilitando que usuário de nível não administrativo possa rodar em seu computador. Todo *Onion Router* executa conexões através do *Onion Proxy*, o proxy da rede *Tor*, para estabelecer conexões na rede e lidar tráfego das aplicações.

Cada roteador possui dois tipos de chaves, uma de longa duração e outra de curta duração. A de longa duração é *identity key* usada para assinar certificados TLS e o *router descriptor*, que é uma listagem das características do roteador. No caso do roteador que é servidor de diretórios ele também vai assinar os diretórios com essa chave.

Já a chave de curta duração, *Onion Key*, serve para decifrar requisições de outros usuários para estabelecer um circuito e criar chaves efêmeras. Esse segundo tipo de chaves é trocado constantemente para evitar danos, caso elas forem comprometidas (DINGLEDINE; MATHEWSON; SYVERSON, 2003, p.3-4).

Para dificultar um atacante modificar os dados ou tentar se passar por um *Onion Router*, os roteadores utilizariam o *Onion Proxy* via TLS e chaves temporárias (DINGLEDINE; MATHEWSON; SYVERSON, 2003,4).

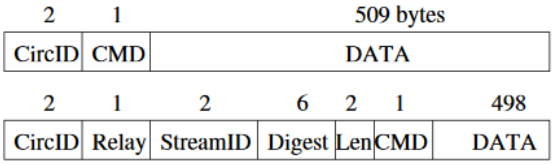
O tráfego é feito por células de tamanho fixo de 512 bytes, que são formadas pelo *header* e o *payload*. No *header* tem o "*circuit identifier*" e o comando para se fazer com *payload*, sendo que a Figura 6 detalha isso graficamente. Os comandos são (DINGLEDINE; MATHEWSON; SYVERSON, 2003,4):

- *padding: keep alive e para link padding*
- *create (d):* criar um novo circuito
- *destroy:* desmantela um circuito

Também, a Figura 6 representa graficamente os *Relay cells* com um *header* adicional, o *relay* header, antes do *payload*, o *StreamID* já que um circuito poderia ter vários *Streams* um *checksum* de ponto-a-ponto para checar a integridade, o comprimento do *payload* e um *relay* comando.

- *Relay data*: para os dados fluírem na *stream*
- *Relay Begin*: para abrir uma *stream*.
- *Relay End*: para fechar uma *stream* de forma correta
- *Relay teardown*: para fechar uma *stream broken*
- *Relay connect*: notifica o *O*nion *Proxy* que o *relay begin* foi um sucesso
- *Relay extend* (ed): expende o circuito por mais um *hop* e para *acknowledge*
- *Relay truncate* (d): desmantela parte do circuito e para *acknowledge*
- *Relay sendme*: usados para controlar o congestionamento
- *Relay drop*: usado para implementar *dummies*.

Figura 6 – Estruturas de uma command *cell* e da *relay cell*



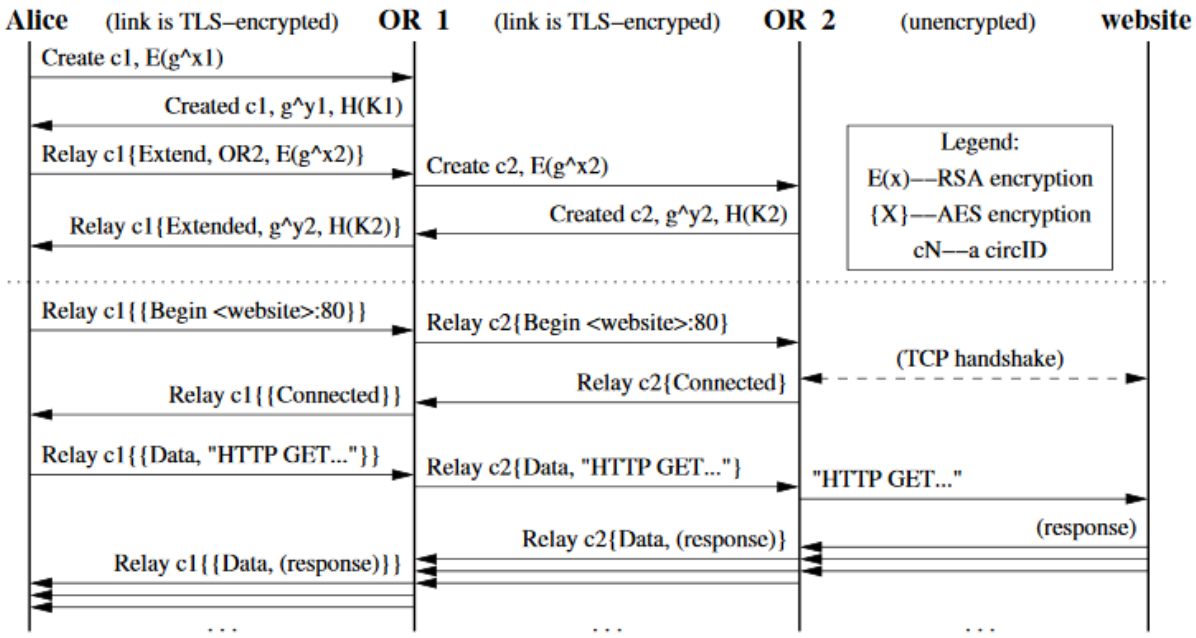
Fonte: DINGLEDINE, MATHEWSON, SYVERSON (2003)

No *Onion Routing* original, cada *stream* de *TCP* terá um único circuito. No *design* do *Tor* cada circuito é usado por vários *streams* facilitando para o *Onion Proxy* se recuperar de falhas de criação de um circuito sem atrapalhar a experiência do usuário por causa da expansão incremental da conexão.

O circuito é construído da seguinte forma: O ponto A envia um comando *create cell* ao nó que fará conexão entre as extremidades dessa parte do circuito, o ponto B. Ponto A escolherá um novo *circID* para ser usado em sua conexão como *Ponto* B e realizará a primeira parte do *Diffie-Hellman handshake* utilizando a *Onion Key* do *Onion Proxy* criando uma conexão criptografada entre ele e o ponto B que respondera com mesmo procedimentos em direção ponto A.

Ponto A enviará um *relay extend cell* para ponto B especificando o próximo onion *relay*, o ponto C. Ponto B copia a parte do *handshake* em um *create cell* e entregará para o ponto C que cria outro circuito e *circID* com o ponto B, sobre o qual o ponto A nunca vai saber. Quando o ponto C responde para ponto B, o mesmo responderá para ponto A colocando no *payload* um *relay extended cell* que irá para ponto A. Na Figura 7 exemplifica como ponto A não tendo o *circID* do circuito entre ponto B e C, mas tem uma troca de *chaves* com ponto C de forma indireta. Para maior expansão é só repetir o processo de forma recursiva. Isso é chamado de *Forward Secrecy e Key Freshness*.

Figura 7 – Exemplo da criação de um circuito com dois nós



Fonte: DINGLEDINE, MATHEWSON, SYVERSON (2003)

Após o estabelecimento do circuito, o ponto A consegue enviar *relay cells*. Ao receber uma o *Onion Relay*, o no receptor vai decrepitar o header e o payload com a chave da sessão do circuito e verificar se a assinatura digital, o digest, é válida. Se for

válida, o processo continua, mas em caso de falha antes do último nó, o *circID* apropriado será corrigido, e caso estiver no último Onion *Relay*, o circuito é desmantelado.

Devido o *digest* ser criptografado com um valor diferente a cada pulo, só fará sentido para o Onion *Relay* específico. Isso é uma topologia leaky Pipe, que permite o Ponto A ter diferentes Onion *Relay*s de saída (a isto se denomina camadas).

Para desmantelar um circuito, o ponto A envia um *destroy control cell*. Isso causa um efeito em cascata para todos os *Onion Relay*, que recebe o comando, executa-o e transmite para o próximo *relay*. O caso o ponto A envie um *truncate cell* para um *Onion Relay* específico, o mesmo vai enviar o comando destroy *cell* para os próximos *Onions Relays*.

Quando uma aplicação em Ponto A quer uma conexão *TCP* ela pede para o *Onion Proxy*, por *SOCKS*, para conectar com um nó escolhendo o circuito mais recente e um *Onion Relay* que tenha política que atenda requisitos para ser o nó de saída daquela conexão. O *Onion Proxy* enviará um *relay connect* que criará randomicamente um *StreamID*. Quando o nó de saída se conecta com o *Host* remoto ele responde com um *relay connected*. Nesse momento o *Onion Proxy* aceitará *stream* de *TCP* da aplicação. Esses dados serão empacotados como *relay data cells* e serão enviados para o próximo nó.

O primeiro modelo de *Onion Routing* era vulnerável a ataques de *malleability* por não usar checagem a integridade do *stream cypher*.

Por usar *TLS* em suas conexões, o *Tor* dificulta adversários externos de modificar o conteúdo transmitido. A checagem de integridade poderia ser feita em cada pulo, mas isso causaria uma expansão no overhead na mensagem e essa solução só funcionaria no tráfego de ponto A para frente. Por fim, no artigo assume que o ataque de ponto a ponto *timining* vai ser uma vulnerabilidade nativa do *Tor*.

É necessário ter em mente que devido a topologia *leaky-pipe*, qualquer pulo no circuito pode ser uma ponta.

Para um atacante ter certeza de que conseguiu remover ou modificar uma *cell* ele precisa deduzir o estado do “*digest”* naquele momento. O artigo explica que o

esforço computacional do “*digest”* se comparado com o uso do *AES* é mínimo, e também é só utiliza 4 *bytes* por *cell* para minimizar a carga enquanto o risco de um adversário adivinhar um *hash* válido é aceitavelmente baixo sendo que o resultado de um *hash* errado é o desmantelamento do circuito.

Outro ponto que explica que para não desgastar os servidores dos voluntários do projeto *Tor*, é usado um algoritmo do tipo *token bucket* para tentar manter uma média durante longo tempo enquanto permite um *burst* de curta duração.

Preocupando-se com os congestionamentos tanto intencionais como acidentais apesar de ter sido planejado uma limitação de tráfego. Isso pode acontecer através de um grupo de usuários escolherem o mesmo nó de entrada e saída para seus circuitos. Na ausência de um controle de congestionamento esse gargalo pode se propagar na rede.

Throttling a nível de circuito: para controlar o uso de banda, cada *Onion Relay* analisa duas *windows*, que são espécie de contadores, a de *packaging* que vê quantas *Relay data cells* o *Onion Relay* pode empacotar para transmitir e a *delivery* que mostra quantos pacotes estão dispostos para entregar.

Cada *window* começa com um certo valor de *data cells,* sendo quando são empacotados ou entregues a *window* decrementa-se. Quando o *Onion Relay* recebeu *data cells* suficiente ele enviará um *relay sendme cell* com o *StreamID* zero, incrementado sua *packing window* com a adição de 100.

Se a janela de empacotamento chega a zero o *Onion Relay* para de ler conexões *TCP* de todas as *streams* do circuito correspondente e ele não enviaria mais *relay data cells* até receber um *sendme cell*.

O *Onion Proxy* trabalha deforma similar sendo que tem que rastrear a *window* de *packaging* e *delivery* para todo *Onion Relay* do circuito. Quando a *window* de empacotamento chegar a zero ele enviaria *stream* para aquele *Onion Relay*.

Throttling a nível *Stream*: É parecido com o do circuito onde o *Onion Proxy* e o *Onion Relay* usam o *relay sendme* para controlar o fluxo de ponto-a-ponto de cada *stream* de window de empacotamento que começará com 500 e seria incrementado

com valor fixo de 50. Além da checagem padrão ele também verificaria se fluxo do *TCP* entregue com sucesso.

Outro recurso descrito no *paper* são os *Rendezvous Points* que atuam em blocos de construção para encontrar *hidden services*, isso permite que um provedor ofereceria um serviço de *TCP* sem revelar o seu *IP* e de forma indireta protegeria contra os ataques de negação de serviço.

Esses *Rendezvous points* tem como objetivos controlar o acesso de modo que os *hidden services* conseguiriam filtrar requisições, garantir robustez da rede sendo que eles conseguiriam de manter um pseudônimo por um longo tempo sem depender de um *Onion Relay* especifico, ser capaz de migrar um serviço através dos *Onion Relays*, *smear resistance* para que um atacante social não conseguisse ser capaz de imputar culpa legal À um *Rendezvous router* oferecendo um serviço ilegal ou fazendo o router parecer ser o culpado e *application transparency* que é permite os serviços funcionarem sem necessitar adaptações,

Para esse funcionamento os *hidden services* escolheriam alguns *Onion Relays* para serem seus *introduction points*. Um requisitante escolhe um *Rendezvous points* e informa para um dos *introduction points* o *Rendezvous points* dele e o segundo vai se conectar com o Rendezvous point do primeiro. Isso ajuda o Ponto B a escolher que requisições selecionar.

O processo ocorre nos seguintes passos. Os *hidden services* geram uma chave pública de longa duração que identificaria seu endereço e escolheria alguns *Introduction Points* e propagaria eles para o *lookup service* e assinaria a propagação com sua chave pública. Um requisitante descobre sobre o endereço do serviço provido e tenta conseguir mais dados através de *lookup* na rede *Tor*. No momento que a conexão do requisitante escolhe um *Onion Relay* para ser seu *Rendezvous* points é entregue um *cookie rendezvous* para ser identificado por ponto B.

O requisitante abre uma conexão anônima passando por um circuito da rede *Tor* para chegar em um *introduction point* de ponto B e informa sobre si mesmo, o *rendezvous point*, e o cookie dele para iniciar um *D.H*. Handshake e um *hash* da *sessions key* que ele agora compartilha.

O R.P. conecta ambos os circuitos, mas não reconhece que transmite os dados ou as informações transmitidas. Dessa forma é estabelecida uma *stream* anônima permitindo o fluo de dados normalmente entre ambas as pontas. É descrito que o *hidden service* deve abrir uma quantidade razoável de *introduction points* para enviar e que esses pontos sejam atacados por meio de um *DoS*.

O servidor seria configurado para utilizar com o *Onion Proxy* através do *SOCKS*. O serviço escolhido utilizaria o *Onion Proxy* sem modificações ou saber que está usando o mesmo. Também geraria um *pseudo-top-level domain* valido com final de *.onion*, cookie de autorização e os *hash* da chave pública.

Apesar das técnicas de controle de fluxo, pelo *Tor* ser um serviço público isso abre um grande leque de opções de ataques contra a rede. Um exemplo citado é que ao invés de realizar um ataque através do consumo de banda de rede seria possível atacar pelo consumo de *CPU* devido ao gasto relativamente alto que operações criptográficas usavam. Isso poderia ser alcançado através de falsas iniciações de *TLS handshake* enquanto o atacante teria um baixo consumo por só estar iniciando a requisição.

Na época do lançamento do artigo os integrantes do Projeto do *Tor* não tinham realizado nenhuma implementação de alguma medida defensiva, porém já tinham ideias para possíveis soluções. Um modo de resolver isso seria através de que o cliente resolvesse um quebra-cabeça para criar conexões *TLS*. Através de um sistema de *token* de autenticidade com limitada amplitude. Adicionalmente poderia criar um imite para o aceitamento de create *cells* e *TLS* connections, porém isso permitiria um atacante limitar a capacidade de usuários legítimos de criarem novos circuitos.

Um atacante também poderia atacar os limites dos *hosts* e da rede. Derrubando um início de circuito ou *link* derrubaria todas as *streams* passando pelo circuito. Até o lançamento do artigo científico, esse ataque era visto como intermitentes falhas de rede. Isso poderia ser resolvido através de um protocolo *end-to-end TCP like acknowledgment*. Através disso só com a queda do ponto de entrada ou saía teria perda das *streams*, porém isso poderia atrapalhar a performance ou o anonimato.

O problema de abusos e vandalismo através de nós de saída, já tinha sido imaginado pelo projeto *Tor* como uma situação para ser mitigada. Como solução que

foi apresentada no artigo seria o *exit policy* como um meio de dar mais opções para quem mantivesse um nó de saída. Como forma mais liberal de uso de um *exit node* é o *open exit nodes* o qual permite conexões de qualquer tipo.

No outro extremo há nós configurados para não trabalhar como *exit nodes* sendo chamados de *middleman* além dos *private exit nodes* qual permite funcionar para um *localhost* exclusivo ou uma rede específica. Porém, maior parte dos nós de saída são *restricted exit* que tem serviços selecionados e excluído outros como o bloqueio de *STMP* que é algo padrão do *restricted exit*.

No lançamento do *paper*, muitos administradores só permitiam o uso de protocolos como *HTTPS*, *SSH* e o *AIM*. Porém mesmo com esses protocolos poderiam sofrer abusados, sendo recomendado no artigo medidas secundarias no nós de saída como o uso do software *spam assassin*.

Outro ponto que foi sugerido era que o Onion *Relay* se apresenta-se como um serviço de anonimato dando o exemplo tendo *hostname* como a palavra *anonymous*.

A mistura de *open* e *restricted nodes* entregaria uma flexibilidade, e um grande número *middleman* garante que seria rede robusta, porém um número reduzido de *exit node* facilitaria para ataques de análise de tráfego. Isso é remediado com o modelo descrito como *hydra* onde há muitas entradas para poucas saídas criando uma dificuldade de saber quem utilizou o *exit node* naquele momento.

Para os *autores* do artigo original do *Tor* a percepção do público e um parâmetro de segurança e a redução na diversidade dos usuários do sistema *Tor* é uma redução do anonimato em si.

2.1.2 *I2P* – THE INVIBLE *INTERNET* PROJECT

Uma das melhores definições sobre *I2P* é dada por um artigo da *IVPN*. É definido que *I2P* é como uma rede dentro da *Internet* que tem como objetivo manter seu tráfego dentro de si mesma ao limitar navegação do usuário, mas protegendo o mesmo de outros perigos. O autor a descreve como que um dia poderá ser uma verdadeira *Darknet*. *(*HOLDEN, [s.d.]).

No artigo acadêmico *I2P -The Invisible Internet Project (*Astolfi, Kroese, van Oorscho) *detalha* como ela foi projetada para funcionar como uma rede paralela dentro da *Internet* que pudesse ser acessível através de navegadores normais e sem ter pontos centrais de comunicação resultando que todos os usuários seriam roteadores de forma compulsória nessa rede. Também é descrito que os usuários teriam a capacidade de customizar as configurações dela para se encaixar com suas necessidades.

Também é descrito como uma camada de *IP* segura e anônima que utiliza *location independente identifiers* no lugar de endereços *IP s* tradicionais para fazer a comunicação. Também é descrito que as *messages*, os substitutos do pacote de *IP*, podem ser significativamente maiores do que os mesmos.

Outro ponto é a utilização de tuneis unidirecionais para o transporte de dados entre os roteadores. Esses tuneis podem ser divididos em *outbound tunnel* e *inbound tunnels*. Os *outbound tunnels* são tuneis que enviam dados para outros roteadores enquanto os *inbounds* recebem dados de outros membros. Todo roteador terá diversos tuneis de entrada e saída.

No mesmo artigo descreve o funcionamento da transmissão de *messages* da seguinte forma. Para ponto B chegar até ponto A ele vai te ter conhecimento de quais roteadores ele precisa conectar para sair dos seus tuneis e quais são os roteadores com tuneis de entrada que conectaram no roteador de Ponto A. Essas informações são adquiridas através do *routerInfo* e *leaseSet*. Outro componente importante é o *Network Database*, *netDb*, que é uma tabela de *hashes*.

A primeira ação que ponto B vai precisar realizar é construir um túnel usando *routerInfo* para essa atividade. No *routerInfo* terá a identidade do roteador, endereço de contato, *IP* e número da porta, data de publicação, um texto exibido opções para ser utilizado em debug e uma assinatura dos dados anteriores. Com esses dados ela vai enviar um comando *build message* para primeira conexão do túnel e instruções para serem passada adiante nos *routers* que serão usados para construção do túnel.

Agora ponto A precisa saber como chegar até o Ponto B. Para isso ele terá que pedir o *leaseSet* de ponto B ao *netDb.* O *leaseSet* contém o gateway do túnel do roteador, O *ID* do túnel e a data de expiração dele, destino em si, uma chave para

criptografia ponto a ponto, uma chave pública que não tinha sido implementada para utilização e assinatura digital de todos esses dados.

Dessa forma ponto A tem o conhecimento de como enviar a *message* até o Gateway do túnel de entrada do Ponto B que vai levar a mensagem até Ponto B. Junto com a *message* o Ponto A pode enviar opcionalmente seu próprio *leaseSet* para o Ponto B não precisar puxar os dados no *netDb* para responder.

Como já foi explicado a rede *I2P* tem ferramentas próprias acessíveis pelo navegador O primeiro exemplo dela é *I2P E-mail* que serve como um serviço de e-mail comum para rede, porém sem a possibilidade de enviar mensagens fora da rede. Devido sua natureza descentralizada em caso de perca da senha usando configuração padrão, é impossível de recuperar o *e-mail*.

Eepsites são equivalentes ao *Websites*. Um ponto importante é a ausência de um serviço de *DNS* centralizado sendo que qualquer usuário pode registrar um domínio *.ip2*. Também é apontado que devido a velocidade da rede no momento do lançamento do artigo, era encorajado a criação de *Eepsites* que usassem pouca banda de *Internet*.

Outra ferramenta é o *torrent* que é mais seguro e anônimo que as *VPNs* oferecidas no mercado. Devido a descentralização da rede garante uma maior resistências aos arquivos transferidos nela e uma dificuldade de aplicar sanções legais contra usuário dela. O maior problema da utilização de *torrents* é a velocidade de transmissão.

Na comunicação entre roteadores é descrita como protegida na camada de transporte. Cada pacote é cifrado com chaves *AES-256/cbc*. As chaves são trocadas em uma troca de chaves *D. Hellman* de 2048 bits. Os *tunnel messages* são criptografados com chaves *AES256/CBC* em explicito IV e verificando no final do túnel com um *SHA* de 256 *bits*.

Outras mensagens são *garlic messages* usando encriptação típica do *I2P*.A diferença do *I2P* adicionar múltiplas mensagens dentro da camada de criptografia do próprio protocolo. As *messages* de *Delivery Status* têm instruções para enviar uma outra *message* confirmando a entrega.

A *Database Store message*: Tem um *Leaseset* para o Destino do originador da mensagem. Isso reduz *requests* para com *netDb* para melhorar a latência e reduzir o risco de ataques de análise de tráfego.

Comparando com o *Tor* é focado para ser um uma espécie de *proxy* para todo tipo de conexão *TCP* na *Internet* com endereços de domínios padrões, em especial na *Web*, a *I2P* ser um jardim murado oferecendo todos os recursos que os usuários poderiam necessitar.

Outro ponto que pode ser interessante é que *I2P* foi desenvolvida principalmente em *Java* enquanto o *Tor* preferiu a utilização da linguagem C. Também é descrito rede *I2P* tem a capacidade nativa de transferência de *torrent* enquanto o *Tor* não tem essa capacidade.

Como pontos fortes, a *I2P,* foi projetada para ser uma rede autossuficiente fazendo quase tudo que é possível por outros programas, mas de forma anônima através aplicações desenvolvidas exclusivamente para essas redes. Os tuneis quem tem um tempo de curto de existência para dificultar ataques. Ao obrigar todos os clientes da rede a trabalharem como roteadores para aumentar o nível de anonimato através dos números.

Entretanto como fraquezas a rede tem poucos *outproxies* para conseguir acessar endereços de domínios comuns, seu *design* é frágil caso a quantidade de usuários for pequena causando limitações no anonimato, além de seu estado experimental e era consideravelmente mais lento que uma conexão normal de *Internet*.

3 TESTE DE REDES SOBREPOSTAS E OUTRAS FERRAMENTAS

Foram realizados três tipos de teste práticos para comparar o *Tor* e *I2P* com outros programas. Esses experimentos foram classificados como teste básicos, testes avançados e os testes de *hidden services*.

Os testes básicos têm como objetivo exibir os resultados das ferramentas sem ter realizado nenhuma interação além das necessárias para seu funcionamento tentando simular a utilização de um usuário leigo. Nessa bateria de testes foi usado os serviços do *IPlocation*, *JonDonym* e *Whoer* para coletar dados em relação às falhas seguranças comuns. Esses serviços foram escolhidos devidos a facilidade de utilização.'

Os testes avançados têm como objetivo utilizar os resultados dos testes do *JonDonym* e *Whoer* para apresentar alguns problemas que pode atrapalhar o anonimato da conexão e mostrar como corrigir essas deficiências dos métodos apresentados.

Esses dois testes foram planejados para serem complementares. Os testes básicos não têm profundidade, mas tem uma vasta variedade de ferramentas analisadas enquanto os avançados demostram-se como tirar o melhor de ferramentas configuradas.

Os testes de *hidden services* é para mostrar a capacidade de utilizar o *Tor* para gerar um website na rede *onion*. Foi realizado uma pequena monitoração da conexão entre clientes e esse servidor.

Cada análise realizada pelos serviços *online* tinha objetivos bem específicos. O primeiro teste utilizou os serviços do *IPlocation,* o qual entrega como um relatório que lista as prováveis localizações do endereço de *IP* utilizado na conexão.

A segunda fase pertence ao *JonDonym* que é encontrado em um endereço externo do *site* da instituição. Possui um problema que de que alguns campos são ocultados caso nenhuma falha seja encontrada para aquele campo específico. Esse foi no caso do *HTTP_X_FORWARDED_FOR* que era exibido em resultados positivos e ocultados em negativos. Devido esses problemas, ele foi utilizado de forma complementar ao teste do *Whoer*.

A parte final e principal dessa bateria de teste é forma expendida da avaliação online do *Whoer* que se encontra na página principal do *site*. Os principais campos escolhidos para serem analisados foram todos do cabeçalho e o campo *WebRTC* no corpo do teste.

Também foram realizadas gravações de conexão através Wireshark para acompanhar a mesma e registrar o funcionamento dos testes.

3.1 TIPOS DE FERRAMENTAS

Foram analisados os resultados dos seguintes recursos nos testes básicos: *proxies*, *JAP*, *VPNs*, *Tor* e o *I2P*.

Proxies foram divididos pelo modo de utilização em dois grupos *web proxy* e *proxies*. O subgrupo dos *proxies* foi subdivido em 3 categorias pelo nível de anonimato transparente, *anonymous* e *elite* e separados pelo modo que foram aplicados na conexão sendo o navegador ou gerenciador de *proxy* do sistema.

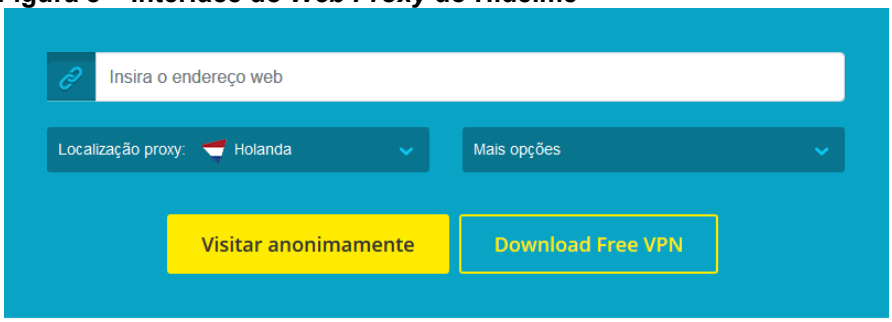
VPNs foram divididas pelo modo de aplicação de seus recursos: extensões dos navegadores, cliente proprietário e o *OpenVPN*.

3.1.1 *WEB PROXY*

Nesse subcapítulo o termo *web proxy* será usado para denominar endereços de *websites* que oferecem através de suas interfaces gráficas um método de utilização dos seus serviços de *proxies*. Entretanto o termo também é utilizado com outros significados pela academia e pela imprensa para definir diversos tipos de serviços e protocolos.

Na matéria da *Lifewire What Is a Web Proxy*? *(*Jerri Collins, 2018) define o modo de utilizar esse tipo de serviço *online* semelhante a um motor de busca. A Figura 8 mostra essa aparência, no qual o usuário digita o endereço de destino no campo do formulário em um *website* para criar uma conexão entre a máquina cliente, o serviço de *proxy* como intermediário e o endereço do servidor *Web*.

Figura 8 – Interface do *Web Proxy* do Hide.me

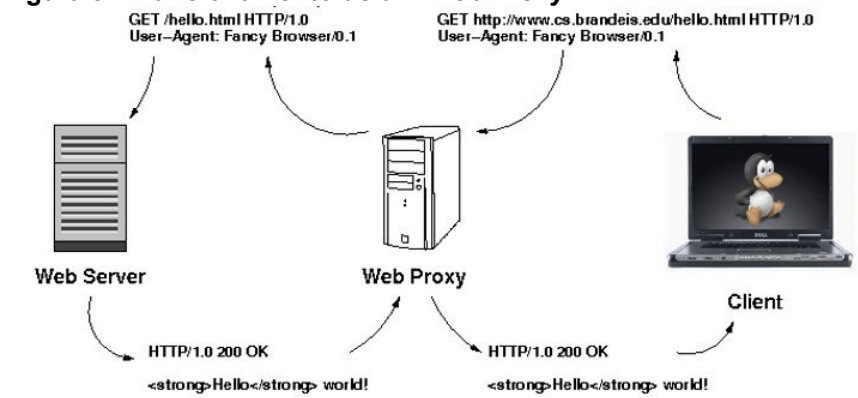


Fonte: Hide.me

3.1.2 *PROXY*

Nesse capítulo, o termo proxy será usado conforme a definição do artigo acadêmico, *Survey Paper on Rising Threats of Subverting Privacy Infrastructure* (Balasuramanian, 2016, p.1) em relação aos endereços IP de servidores e suas respectivas portas que são utilizadas para intermediação de uma ou mais conexões dos clientes e os endereços de destino sendo representado graficamente na Figura 9.

Figura 9 – Funcionamento de um *Web Proxy*



Fonte: PPLware

Foram escolhidas duas formas de aplicar os serviços de *proxies* selecionados para os testes. A primeira foi através da configuração direta e exclusiva no navegador *Firefox* o endereço de *IP* do servidor *proxy* e sua respectiva porta. Infelizmente não foi possível replicar em outros navegadores devido limitações desses softwares sem a utilizações de extensões.

A segunda foi através do componente de gerenciamento de *proxies* do sistema operacional. Isso faz que todos os navegadores e programas configurados serem gerenciados por esse componente do sistema operacional utilize a conexão através *proxy* configurado.

A Nova *Proxy* foi escolhida como fonte dos endereços que oferecem serviços gratuito de *proxy* devido ser um dos poucos casos que explicava a divergência técnica entre os níveis de anonimato dos três tipos de classificações para os *proxies* que ela definiu em na matéria *Proxy Anonymity Levels (*Nova *Proxy*, [s.d]) . Segundo essa empresa, existem três níveis de anonimatos que um *proxy* pode ser classificado: o transparente, o *anonymous* e o *elite*.

As denominações foram escolhidas pela própria empresa e na matéria a mesma explica que a nomenclatura pode variar um pouco de uma fonte para outra. como o termo "*High Anonymous"* que é utilizado em no *List Free Proxy* um *site* que também cataloga serviços de *proxies gratuitos*.

A diferença entre esses três tipos de *proxies* se encontra no já citado campo do cabeçalho de *HTTP*, *HTTP_X_FORWARDED_FOR*. O tipo transparente transmite o endereço de *IP* fornecido pela *ISP* do utilizador do serviço nesse campo. O tipo *anonymous* não transmite nada no campo, porém não tenta esconder o fato da conexão ser feita através de um *proxy* tendo o tipo do servidor utilizado na conexão no campo do cabeçalho *HTTP_VIA*. O tipo *elite* não transmite nada em ambos os campos e esconde o fato ser um *proxy*.

Outro ponto importante na escolha de serviço *proxy além* do grau de anonimato que o serviço alega possuir, é a capacidade de utilização dos protocolos de segurança que permitem a utilização de uma conexão *HTTPS*. Alguns serviços de *proxies* dos três graus de anonimato só têm capacidade de realizar conexões em *HTTP* o que significa na transmissão dos dados de forma não criptografada.

Por é necessário lembrar que serviços gratuitos de *proxies* não garante que suas fontes são seguradas. A Nova *Proxy* diz no topo da página de pesquisa dos *proxies* que o *software* dele verifica os servidores *proxies* disponíveis enquanto a free-*proxy*.cz é mais direta dizendo que utilização do banco de dados deles é por risco do usuário e que eles têm nenhum controle sobre o os servidores de *proxy* oferecidos.

3.1.3 *JAP*

Balasuramanian (p.2) explica que o JAP é um mix cascades de baixa latência que permite que o usuário escolha qual cascade ele pode usar. Em 2012 ele era o

segundo em popularidade atrás do Tor. Comercialmente o JAP é chamado de *JonDonym*. Shahbar e Zineir-Heywood (p.4) explicam que os serviços gratuitos têm só duas etapas enquanto os pagos têm três.

3.1.4 *VPNs*

Nos testes foi escolhido utilizar três métodos de avaliar o uso das VPNs. O primeiro método foi através de extensões no navegador, o segundo foi através de um cliente proprietário e o terceiro foi através da utilização do *OpenVPN*.

3.1.5 REDES SOBREPOSTAS

As duas *redes sobrepostas* que serão utilizadas nos testes. O *Tor* e *I2P*. Porém, como já foi explicado no capítulo sobre ambos, suas características de *Darknet* não podem ser testadas nos testes básicos e avançados. *I2P* foi avaliado pelos seus resultados do *outproxy*. Já o *Tor* foi testado em cinco formas diferentes. Inicialmente foi configurado para rodar direto do *Firefox* através de suas configurações de *proxy*. A segunda forma foi através da configuração em “Configuração Manual de *Proxy*” no painel de controle do *Windows*.

A terceira linha de teste foi a utilização do *Brave* e o serviço do *Tor* embarcado. O quarto foi através do *Jondofox*. O quinto teste foi através do *Tor Browser Bundle*, método aconselhado pelo *Tor Project* com o meio mais adequado para utilizar o *Tor* para navegação usando a configuração de segurança mediana, sendo a única etapa dos testes que teve uma configuração extra além da necessária do que o recurso de ocultação precisa para funcionar.

3.2 TESTES BÁSICOS

3.2.1 TESTE COM *WEB PROXY*

Foram utilizados os serviços de web *proxy* do *Hideme*, *hidester* e o *ProxySite* para serem testados nos navegadores *Google Chrome, Firefox e Edge*. Os serviços foram escolhidos devido a uma publicação de uma matéria por Bradley Mitchell no *website lifewire* tendo uma lista organizada que explana qualidades e defeitos que cada serviços tem.

Não foi possível realizar os testes do *IPlocation*(*iplocation.net*), por causa de alguma configuração interna que esses serviços possuem sendo substituído pelo *iplocation*.com nessa ocasião.

Usando o *Hideme* problemas foram encontrados como teste *JonDonym* em que todos os navegadores tiveram seus *IPs* locais vazados, sendo o *Firefox* por um erro de *HTTPS* e os outros dois devido a problemas no *JavaScript* como consta na Tabela 1.

Tabela 1 - *Web Proxy – Hideme*

Web Proxy (hideme)			
Navegadores	Firefox	*Google Chrome*	*Edge*
Localidade do *Proxy*			
IPLocation.com	Alemanha	Alemanha	Falha
JonDonym			
Ip-check.info	Serviço de *proxy* e vulnerabilidade de *HTTP*S detectadas	Serviço de proxy e vulnerabilidade de *JavaScript* detectadas	Serviço de proxy e vulnerabilidade de *JavaScript* detectadas e resultado sem formatação
Whoer			
Proxy	Não	Não	Falha
Anonymizer	Não	Não	Falha
WebRTC	Não	Não	Falha
DNS	Não	Não	Falha
Problemas Secundários	Sim	Sim	Falha

Fonte: Próprio autor

Usando o *Hidester* houve problemas entre o *Edge* e o teste o *IPlocation.com*, porém os outros dois navegadores completaram o teste e indicaram um *IP* Canadense nesse teste. No segundo teste novamente o *Edge* apresentou uma falha na execução e os outros dois navegadores vazaram o IP da máquina local por *JavaScript*.

Na Tabela 2 mostra que o último teste o *Chrome* apresentou problemas com o funcionamento do teste e os outros dois não foram detectados como *Proxy* ou

Anonymizer nos testes principais, mas os testes secundários os taxaram de *Web Proxies*

Tabela 2 - *Web Proxy – hidester*

Web Proxy (hidester)			
Navegadores	Firefox	*Google Chrome*	*Edge*
Localidade do Proxy			
iplocation.com	Canada	Canada	Falha
JonDonym			
Ip-check.info	Serviço de *proxy* e vulnerabilidade de JavaScript detectadas	Serviço de *proxy* e vulnerabilidade de JavaScript detectadas	Falha
Whoer			
Proxy	Não	Falha	Não
Anonymizer	Não	Falha	Não
WebRTC	Não	Falha	Não
DNS	Não	Falha	Não
Problemas Secundários	Sim	Falha	Sim

Fonte: Próprio autor

Na Tabela 3 consta os testes usando o *web proxy* da *ProxySite* que funcionou no teste localização para o *Firefox* e o *Chrome* apontando um endereço americano. No *JonDonym* todos os navegadores vazaram o *IP* público do computador que realizava o teste. No *Whoer* mostrou vazamento do *IP* público pelo *Firefox* por causa do *WebRTC* e os testes secundários indicaram o *Web Proxy*.

Tabela 3 - *Web Proxy - ProxySite*

WebProxy (Proxy Site)			
Navegadores	Firefox	*Google Chrome*	*Edge*
Localidade do *Proxy*			
iplocation.com	Estados Unidos	Estados Unidos	Falha
JonDonym			
Ip-check.info	Serviço de *proxy* e vulnerabilidade de *HTTPS* detectadas	Serviço de *proxy* e vulnerabilidade de *JavaScript* detectadas	Serviço de *proxy* e vulnerabilidade de *JavaScript* detectadas
Whoer			
Proxy	Não	Não	Não
Anonymizer	Não	Não	Não
WebRTC	Sim	Não	Não
DNS	Não		
Problemas Secundários	Sim	Sim	Sim

Fonte: Próprio autor

3.2.2 TESTE COM *PROXY*

Foram escolhidos serviços de *proxy* gratuitos de uma lista dinâmica no *website* da empresa *Nova Proxy* devido explicarem a diferença entre os níveis de segurança dos *proxies* oferecidos. Também foram adicionados aos testes os *proxies* do *free-proxy.cz* pela sua velocidade.

Um dos critérios para seleção dos *proxies* utilizados nos testes foi o suporte para conexões *HTTPS*. Infelizmente a lista da *Nova Proxy* não exibe quais endereços tem a capacidade de fazer uma conexão segura sendo necessário tentar adivinhar pelo número da porta e posteriormente testar a conexão criptografada manualmente.

3.2.2.1 TESTES COM *PROXY* TRANSPARENTE

Foram escolhidos para o teste de *proxy* transparente dois *IP* da *Proxy Nova* sendo dois deles brasileiro (168.227.213.83 para teste com navegador e posteriormente 200.178.251.146 na configuração do sistema) e outro americano *(*157.245.224.29) e um endereço americano do *Free-Proxy.cz (*38.91.107.242*).* Os três trabalharam utilizando a porta 8080.

Inicialmente foi configurado o *proxy* no navegador *Firefox*. O teste com o *IPlocation* mostrou as prováveis localidades dos endereços de *IP*. O *IP* brasileiro indicou cidades diferentes nas três análises enquanto o serviço da *Free-Proxy* mostrou duas localidades diferentes sendo pelo *DB-IP* mostrava uma posição mais específica.

No *JonDonym* o *IP* brasileiro foi detectado como um *Proxy* e mostrou o *IP* válido no campo *X-FORWARDED-FOR*, um tipo de servidor *proxy* no campo *VIA*. O mesmo aconteceu com os *IPs* estrangeiros sendo que usavam o *software Squid* como servidor, porém em versões diferentes. No teste da *Whoer* o *proxy* nacional não foi detectado, mas apresentou um problema secundário do horário da região do servidor *proxy* não ser a mesma do que o horário na máquina de teste. Outro problema foi o vazamento por *WebRTC* que aconteceu em todas as ocasiões liberado o endereço de *IP* da máquina de teste que tinha na *LAN* e o válido na *Internet*.

A Tabela 4 nos mostra os *IPs* estrangeiros foram indicados como "*maybe*" o qual indica a possibilidade ao invés de uma certeza de que conexão seria um *proxy* apesar da mesma ser transparente.

Tabela 4 - *Proxies* transparentes configurado no navegador

***Firefox* (Transparente *Proxy*)**			
Provedores dos *proxies*	Brasil	Nova *Proxy*	CZ
Localidade do *Proxy*			
I2PLocation	Águas Vermelhas	Santa Clara	*Miami*
ipinfo.io	Pedra Azul	Santa Clara	*Newark*
DB-IP	Rio de Janeiro	Santa Clara	*Newark (Central Business District)*
JonDonym			
Your IP	*Proxy*	*Proxy*	*Proxy*
HTTP_X_FORWARDED_FOR	*IP* Público	*IP* Público	*IP* Público
HTTP-VIA	*Mikrotik*	*Squid (*3.5.27)	*Squid (*3.3.8)
Whoer			
Proxy	Não	*Maybe*	*Maybe*
Anonymizer	Não	Não	Não
WebRTC	Sim	Sim	Sim
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

Nos testes o *proxy* brasileiro foi ajustado em "*Configuração de proxy manual*" para funcionar em qualquer navegador do sistema operacional. Esse procedimento se repetiu para todos os *proxies* utilizados para atingir o S.O. como um todo.

Na Tabela 5 mostra teste da localização do *proxy* apontou São Paulo duas vezes e Goiânia uma vez e todos os testes do *JonDonym* teve o envio do *IP* válido no campo *HTTP_X_FORWARDED_FOR*, o *proxy* foi detectado e indiciou a utilização de uma versão do *Squid* pelo servidor apesar de não exibir uma versão específica a declarando com valor "1.1 *proxy-obj.local* (*squid*)".

No teste do *Whoer* todos os navegadores não foram detectados como *Proxy* ou *Anonymizer* sendo um comportamento não esperando. No *Firefox* houve vazamento de *WebRTC* enquanto os navegadores de base *Chromium* isso não ocorreu. Todos os navegadores tiveram problemas secundários.

Tabela 5 – *Proxy* brasileiro transparente configurado no sistema

Configuração Manual de *Proxy*(*Anonymous*-Brasil)			
Navegadores	*Firefox*	*Google Chrome*	*Edge*
Localidade do *Proxy*			
I2PLocation	São Paulo	São Paulo	São Paulo
ipinfo.io	Goiânia	Goiânia	Goiânia
DB-IP	São Paulo	São Paulo	São Paulo
JonDonym			
Your IP	*Proxy*	*Proxy*	*Proxy*
HTTP_X_FORWARDED_FOR	*IP* Público	*IP* Público	*IP* Público
HTTP-VIA	squid	squid	squid
Whoer			
Proxy	Não	Não	Não
Anonymizer	Não	Não	Não
WebRTC	Sim	Não	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

Na utilização do *proxy* transparente da *Nova Proxy* o *Google Chrome* não conseguiu realizar o teste do *IPlocation*. Já os outros dois navegadores indicaram a cidade de Santa Clara como origem.

Na Tabela 6 o teste do *JonDonym* os três navegadores tiveram o resultado idênticos nos campos avaliados tendo uma versão claro do *Squid* sendo indicada, o *IP* válido e o *proxy* detectado.

Na avaliação do *Whoer* os navegadores *Firefox* e *Edge* foram indicados com possibilidade de que seriam *proxies* pela sinalização *Maybe* no campo que examina o nível da chance do uso de *proxy* ou não. Esse comportamento é mais esperado para um *proxy* transparente do que sua não detecção como nos testes anteriores. Problemas secundários como diferença de horário que o sistema operacional e horário que a conexão de *proxy* tem e a diferença entre o idioma no sistema e o idioma que seria esperado pela origem da conexão foram indicados.

Tabela 6 - *Proxy* transparente da *Nova Proxy* configurado no sistema

Configuração Manual de *Proxy*(*Anonymous-Nova*)			
Navegadores	*Firefox*	*Google Chrome*	*Edge*
Localidade do *Proxy*			
I2PLocation	Santa Clara	Falha	Santa Clara
ipinfo.io	Santa Clara	Falha	Santa Clara
DB-IP	Santa Clara	Falha	Santa Clara
JonDonym			
Your IP	*Proxy*	*Proxy*	*Proxy*
HTTP_X_FORWARDED_FOR	*IP* Público	*IP* Público	*IP* Público
HTTP-VIA	*Squid* (3.5.27)	*Squid* (3.5.27)	*Squid* (3.5.27)
Whoer			
Proxy	*Maybe*	Falha	*Maybe*
Anonymizer	Não	Não	Não
WebRTC	Sim	Não	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

Somente o Firefox foi capaz de gerar um uma página legível para as prováveis posições geográficas das conexões do *proxy* transparente da *CZ,* sendo que os navegadores de base *Chromium* executaram os testes, mas geram uma página inelegível. Já o *JonDonym* teve resultado semelhante ao serviço da *Nova Proxy* só mudando a versão d servidor *Squid* em relação as três bases avaliadas.

Já no Whoer todos os três navegadores funcionaram perfeitamente tendo mesmo resultados com exceção do campo *WebRTC* onde o Firefox exibiu o IP de *LAN* e o IP externo como a Tabela 7 aponta. Todos eles indicaram que talvez fosse *proxy* e tiveram problemas secundários.

Tabela 7 - *Proxy* transparente da *CZ* configurado no sistema

Configuração Manual de *Proxy*(Transparente-*CZ*)			
	Firefox	*Google Chrome*	*Edge*
Localidade do *Proxy*			
I2PLocation	*Miami*	Falha	Falha
ipinfo.io	*Newark*	Falha	Falha
DB-IP	*Newark (Central Business District)*	Falha	Falha
JonDonym			
Your IP	*Proxy*	*Proxy*	*Proxy*
HTTP_X_FORWARDED_FOR	*IP* Público	*IP* Público	*IP* Público
HTTP-VIA	*Squid (*3.3.8)	*Squid (*3.3.8)	*Squid (*3.3.8)
Whoer			
Proxy	*Maybe*	*Maybe*	*Maybe*
Anonymizer	Não	Não	Não
WebRTC	Sim	Não	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

3.2.2.2 TESTE COM *PROXY ANONYMOUS*

Foram escolhidos para o teste de *proxy Anonymous* dois *IP* da *Proxy Nova* sendo dois deles brasileiros *(*138.122.140.35 para navegador e 177.46.148.138 no sistema) e outro americano *(*190.103.178.8) sendo que ambos trabalhavam na porta 8080 e um endereço alemão do *Free-Proxy.cz (*195.4.165.127) que utilizava a porta 3120.

Como os *proxies* configurados diretamente no *Firefox* houve vazamentos de *WebRTC* em todos os casos.

Na questão das localidades, o serviço brasileiro apresentou duas cidades diferentes no norte do país, o americano apresentou uma cidade nos *EUA* chamada de *Pompano Beach* e a cidade argentina de Córdoba. Já o *proxy* da *CZ* indicou três cidades alemães diferentes.

No segundo teste o recurso da *CZ* apresentou falha em executar o teste. Na Tabela 8 mostra que o *proxy* nacional indicou a utilização do servidor de *proxy* embarcado *RouterOS* da *Mikrotik* e o internacional da *Nova Proxy* não retornou nada nesse campo. Ambos foram detectados como *proxies.*

Também na Tabela 8 na etapa de checagem do Whoer, ambos serviços da *Nova Proxy* foram detectados como *PROXY* no campo *Anonymizer* enquanto o da *CZ* não apresentou nenhum resultado positivo tanto no campo *Proxy* com no campo *Anonymizer.* Todos os testes também apresentaram problemas secundários.

Tabela 8 - *Proxies anonymous* configurado no navegador

***Firefox* (*Anonymous Proxy*)**			
Provedores dos *proxies*	Brasil	Nova *Proxy*	CZ
IPlocation			
I2PLocation	Cacoal	*Pompano Beach*	*Hellhorst*
ipinfo.io	Cacoal	*Córdoba*	*Stuttgart*
DB-IP	Pimenta Bueno(Pioneiros)	*Pompano Beach*	*Hullhorst (Schnahorst)*
JonDonym			
Your IP	Sim	Sim	Falha
HTTP_X_FORWARDED_FOR	Não	Não	Falha
HTTP-VIA	*Mikrotik*	Falha	Falha
Whoer			
Proxy	Não	Não	Não
Anonymizer	Sim	Sim	Não
WebRTC	Sim	Sim	Sim
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

Na bateria de testes utilizando um *proxy* do nível *anonymous* brasileiro através do recurso de gerenciamento nativo do *Windows*, todos os navegadores em todas as avaliações apontaram a cidade de São Paulo. Em oposição, todos os navegadores falharam em realizar o teste do *Whoer* tendo uma mensagem do *Cisco Umbrella* relatando a filtragem do conteúdo do *site* por ser classificado "*Proxy/Anonymizer*".

Nos exames realizados pelo *JonDonym* tiveram graves problemas de formatação na exibição de resultados. É denotado na Tabela 9 que todos os *browsers*

foram detectados como *Proxy*, porém no campo *HTTP_X_FORWARDED_FOR* não indicou o valor do *IP* válido do computador de teste, sendo o endereço de *IP* 177.46.148.141. No campo *HTTP-VIA* indicou o uso de *software* de *proxy* que aparenta vir com *MikroTik RouterOS* novamente.

Tabela 9 - *Proxy anonymous* brasileiro configurado no sistema

Configuração Manual de *Proxy*(*Anonymous*-Brasil)			
Navegadores	*Firefox*	*Google Chrome*	*Edge*
Localidade do *Proxy*			
I2PLocation	São Paulo	São Paulo	São Paulo
ipinfo.io	São Paulo	São Paulo	São Paulo
DB-IP	São Paulo	São Paulo	São Paulo
JonDonym			
Your IP	Sim	Sim	Sim
HTTP_X_FORWARDED_FOR	Não	Não	Não
HTTP-VIA	*Mikrotik*	*Mikrotik*	*Mikrotik*

Fonte: Próprio autor

Já como teste com endereço estrangeiro configurado no sistema operacional. os valores das localidades deram dois resultados sendo o primeiro em *Pompano Beach* no EUA enquanto outro apontava para Córdoba na Argentina.

No teste do *JonDonym* os resultados demostrados na Tabela 10 retornaram com os *proxies* detectados porém sem passarem informações sobre a conexão da rede local com os mesmos e sem valores que indicavam o programa que executava a transmissão dos dados.

Na realização da terceira parte ocorreu novamente o *Firefox* vazando *IP* por *WebRTC*. Os três navegadores foram indicados por usarem algum *Anonymizer* que foi taxado de *"PROXY*" enquanto o campo destinado para *proxies* apresentou um valor negativo e todos apresentaram de problemas secundários.

Tabela 10 - *Proxy anonymous* da *Nova Proxy* configurado no sistema

Configuração Manual de *Proxy*(*Anonymous-Nova*)			
Navegadores	*Firefox*	*Google Chrome*	*Edge*
Localidade do *Proxy*			
I2PLocation	*Pompano Beach*	*Pompano Beach*	*Pompano Beach*
ipinfo.io	Córdoba	Córdoba	Córdoba
DB-IP	*Pompano Beach*	*Pompano Beach*	*Pompano Beach*
JonDonym			
Your IP	Sim	Sim	Sim
HTTP_X_FORWARDED_FOR	Não	Não	Não
HTTP-VIA	Falha	Falha	Falha
Whoer			
Proxy	Não	Não	Não
Anonymizer	Sim	Sim	Sim
WebRTC	Sim	Não	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

Com o *proxy* da *CZ* houve a apresentação de três localidades diferente, todas nas Alemanha. Na segunda parte apresentou uma falha completa na realização do teste com *Firefox*.

Nos outros *browsers*, não tinha nenhum valor no campo *HTTP_X_FORWARDED_FOR* sendo esse resultado algo já esperado em qualquer teste de *proxy anonymous*.

Na Tabela 11 é salientado que além disso o dado em *HTTP-VIA* mostrou que era o *Squid* sendo usado, mas não apresentava uma versão clara do mesmo sendo "*1.1 firewall.agbeef.de*" o valor bruto.

O *Google Chrome* apresentou falhou em mostrar os resultados dos testes de *proxy e Anonymizer* além de teve do problema padrão do *Firefox*. Não resultou nenhum resultado positivo para detecção dos serviços da *CZ*, porém ocorreu problemas secundários em todos os navegadores.

Tabela 11 - *Proxy anonymous* da *CZ* configurado no sistema

Configuração Manual de *Proxy* (*Anonymous-CZ*)			
Navegadores	*Firefox*	*Google Chrome*	*Edge*
Localidade do *Proxy*			
I2PLocation	*Hellhorst*	*Hellhorst*	*Hellhorst*
ipinfo.io	*Stuttgart*	*Stuttgart*	*Stuttgart*
DB-IP	*Hullhorst(Schnahorst)*	*Hullhorst(Schnahorst)*	*Hullhorst(Schnahorst)*
JonDonym			
Your IP	Falha	*Proxy*	*Proxy*
HTTP_X_FORWARDED_FOR	Falha	Falha	Falha
HTTP-VIA	Falha	*Squid*	*Squid*
Whoer			
Proxy	Não	Falha	Não
Anonymizer	Não	Falha	Não
WebRTC	Sim	Não	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

3.2.2.3 TESTE COM *PROXY ELITE*

Foram escolhidos para o teste de *proxy Elite* dois *IP* da *Proxy Nova* sendo um deles brasileiro *(*177.91.111.233) e outro americano *(67.205.177.112*) sendo que ambos trabalham na porta 8080 e um endereço francês do *Free-Proxy.cz (*51.28.71.101) que utiliza a porta 3120.

A Tabela 12 mostra os testes realizados no *JonDonym* não trouxeram qualquer resultado diferente em todos os tipos de testes realizados com os *elite proxies*.

Os *proxies elite* da *Nova Proxy* apresentaram a mesma cidade todas as vezes tanto para o nacional quanto para o Internacional. Já o da *CZ* ficou divido entre um endereço francês e dois no Reino unido.

Nos testes do *Whoer* existiram vazamentos de *WebRTC* com todos os *proxies* configurados no navegador. Fora esse problema o serviço com endereço brasileiro não sofreu de nenhuma complicação, porém isso não se aplica para os estrangeiros. Em ambos os casos o campo *Anonymizer* marcou *PROXY* além de apresentar problemas secundários.

Tabela 12 - *Proxies elite* configurado no navegador

***Firefox* (*Elite Proxy*)**			
Provedores dos *proxies*	Brasil	Nova *Proxy*	CZ
Localidade do *Proxy*			
I2PLocation	Bom Jesus da Lapa	*North Bergen*	*London*
ipinfo.io	Bom Jesus da Lapa	*North Bergen*	*Roubaix*
DB-IP	Bom Jesus da Lapa	*North Bergen*	*Purfleet*
JonDonym			
Your IP	Não	Não	Não
Whoer			
Proxy	Não	Não	Não
Anonymizer	Não	Sim	Sim
WebRTC	Sim	Sim	Sim
Problemas secundários	Não	Sim	Sim

Fonte: Próprio autor

Na Tabela 13 no caso do *IP* Brasileiro sendo gerenciado pelo *Windows* apresentou alguns resultados diferente de quando foi configurado direto no navegador, entre tanto a localidade foi a mesma em todos os casos repetindo com teste do *browser*.

Tabela 13 - *Proxy elite* brasileiro configurado no sistema

Configuração Manual de *Proxy*(*ELITE-Brasil*)			
Navegadores	*Firefox*	*Google Chrome*	*Edge*
Localidade do *Proxy*			
Todos os testes	Bom Jesus da Lapa	Bom Jesus da Lapa	Bom Jesus da Lapa
JonDonym			
Your IP	Não	Não	Não
Whoer			
Proxy	*Maybe*	Não	Não
Anonymizer	Não	Não	Não
WebRTC	Sim	Não	Não
Problemas secundários	Não	Não	Sim

Fonte: Próprio autor

Com *Firefox*, a opção de *proxy* marcou como *Maybe*, uma diferença em comparação com o teste anterior. O *Edge* teve problemas secundários e o *Google Chrome* não apresentou nenhum problema.

Na Tabela 14 todas as localidades apontaram para a mesma cidade utilizando o serviço americano da *Nova Proxy* no *IPlocation*.No *Whoer* o *Google Chrome* falhou em apresentar os resultados dos testes de *Proxy* ou *Anonymizer* e teve problemas secundários. Tanto o *Firefox* como o *Edge* foram marcados no teste de *Anonymizer* como *PROXY* também apresentaram problemas menores.

Tabela 14 - *Proxy elite* da *Nova Proxy* configurado no sistema

Configuração Manual de *Proxy* (*ELITE-Nova Proxy*)			
Navegadores	*Firefox*	*Google Chrome*	*Edge*
Localidade do *Proxy*			
Todos os testes	*North Bergen*	*North Bergen*	*North Bergen*
JonDonym			
Your IP	Não	Não	Não
Whoer			
Proxy	Não	Falha	Não
Anonymizer	Sim	Falha	Sim
WebRTC	Sim	Não.	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

Na Tabela 15 mostra que utilizando o *proxy* da *CZ* recorreu os três locais diferentes sendo um francês e dois britânicos.

A avaliação do Whoer apresentou todos os navegadores tendo sido taxados de *PROXY* no campo *Anonymizer* além de ter apresentados alguns problemas secundários.

Tabela 15 - *Proxy elite* da *CZ* configurado no sistema

Configuração Manual de *Proxy*(*ELITE-CZ*)			
Navegadores	*Firefox*	*Google Chrome*	*Edge*
Localidade do *Proxy*			
I2PLocation	*London*	*London*	*London*
ipinfo.io	*Roubaix*	*Roubaix*	*Roubaix*
DB-IP	*Purfleet*	*Purfleet*	*Purfleet*
JonDonym			
Your IP	Não	Não	Não
Whoer			
Proxy	Não	Não	Não
Anonymizer	Sim	Sim	Sim
WebRTC	Sim	Não	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

3.2.3 TESTE COM *JAP*

O primeiro tipo de conexão que se utilizou foi a opção *JAP,* localizado na Alemanha, em Dresden, sendo o endereço de *IP* da saída da conexão 141.76.46.137, que era uma conexão de um salto entre o usuário e o destino.

O teste com o *IPlocation* resultou com o endereço de Dresden, sendo o *DB-IP* apresentando uma região específica da cidade. No teste do *JonDonym*, obteve-se uma avaliação positiva em diversos pontos por utilizar o navegador e *JAP*, mas apresentou um problema no campo *X-FORWARDED-FOR* de forma similar aos resultados de um *proxy* da classe *anonymous* como pode ser visto na Tabela 16. No teste do *Whoer* só houve problemas secundários.

Tabela 16 - *JondoFox – Proxy* JAP

Jondofox (JAP)	
Localidade do *Proxy*	
I2PLocation	*Dresden*
ipinfo.io	*Not Available*
DB-IP	*Dresden (südvorstadt)*
Whoer	
Proxy	Não
Anonymizer	Não
WebRTC	Não
Problemas secundários	Sim

Fonte: Próprio autor

O segundo teste do *JAP* ocorreu utilizando-se uma opção de conexão dupla entre *SpeedPartner* e *Cyrax*, sendo a primeira alemã e a segunda francesa, que é o nó de saída que tem o número de *IP* 178.33.255.188.

O teste com o *IPlocation* executou-se em dois testes, que indicou como Roubaix e um como Watterelos. No teste do *JonDonym* o resultado foi similar ao primeiro, porém sem problema de *X-FORWARDED-FOR*. Na Tabela 17 na parte do teste do *Whoer* foi indicado como um *Anonymizer* e teve problemas secundários.

Tabela 17 - *JondoFox* – Proxy Cyrax

Localidade do *Proxy*	
I2PLocation	Roubaix
ipinfo.io	Roubaix
DB-IP	Gavelines
Whoer	
Proxy	Não
Anonymizer	Sim
WebRTC	Não
Problemas secundários	Sim

Fonte: Próprio autor

3.2.4 TESTE COM *VPNS*

As *VPNs* utilizadas nos testes podem ser divididas em 3 tipos. As que são utilizadas através extensões nos navegadores, as que funcionaram por um cliente proprietário e a que foi configurada através do programa *OpenVPN*.

3.2.4.1 EXTENSÕES DE NAVEGADORES

O primeiro método de teste de *VPNs* que foi utilizando as extensões *ZenMate* e *Hola* nos navegadores. Nesse caso cada extensão fazia uma conexão independente, diferente do modelo centralizado no caso do uso do gerenciador de *proxy*. Como *ZenMate* só foi possível configurar o mesmo para funcionar com navegador *Firefox* e *Google Chrome* enquanto o *Hola* permitiu os três navegadores.

No teste com *ZenMate* selecionado o Brasil como ponto de saída no Brasil. Todos os *IPs* foram marcados como de São Paulo.

Na Tabela 18 é salientado que o *Google Chrome* não exibiu corretamente os resultados do teste de *Proxy* e *Anonymizer* e o *Firefox* recebeu o resultado de positivo no campo de *Anonymizer*, *WebRTC*.e problemas secundários

Tabela 18 - *ZenMate* configurado nos navegadores com *IP* brasileiro

ZenMate (Brasil)		
Navegador	*Firefox*	*Google Chrome*
Endereço de *IP*	181.41.203.97	181.41.203.97
Localidade da *VPN*		
I2PLocation	São Paulo	São Paulo
ipinfo.io	São Paulo	São Paulo
DB-IP	São Paulo	São Paulo
Whoer		
Proxy	Não	Falha
Anonymizer	Sim	Falha
WebRTC	Sim	Não
Problemas secundários	Sim	Sim

Fonte: Próprio autor

A Albânia foi escolhida como segunda opção de localização ao utilizar os serviços da *ZenMate*. Os resultados dos testes do *IPlocation* em ambos os navegadores marcaram em Tirana, a capital do país.

No teste do *JonDonym* a *VPN* foi detectada como um *proxy* enviando valores no campo de *X-FORWARDED-FOR* e *HTTP-VIA* como se estivesse trabalhando igual a um *proxy* transparente. Também estava marcado que utilizava o software Squid, como é indicado pela Tabela 19.

No *Whoer* apresentou a indicação que o *Firefox* estava utilizando os serviços da *Cyber Ghost VPN*, o que deve ter sido uma interpretação errada da página. O navegador do *Google* apresentou problemas de exibir duas primeiras partes do teste novamente. Erros secundários aconteceram em ambos os *browsers*.

Tabela 19 - *ZenMate* configurado nos navegadores com *IP* albanês

ZenMate (Albania)		
Navegadores	*Firefox*	*Google Chrome*
Endereço de *IP*	31.171.152.140	31.171.152.99
Localidade da *VPN*		
I2PLocation	*Tirana*	*Tirana*
ipinfo.io	*Tirana*	*Tirana*
DB-IP	*Tirana*	*Tirana*
JonDonym		
Your IP	Sim	Sim
X-FORWARDED-FOR	Sim	Sim
VIA	Squid/4.8	Squid/4.8
Whoer		
Proxy	Não	Falha
Anonymizer	Sim	Falha
WebRTC	Sim	Não
Problemas secundários	Sim	Sim

Fonte: Próprio autor

Na Tabela 20 Os testes de localização apontaram a origem dos endereços de IP do *Hola* para São Paulo

Tabela 20 - *Hola* configurado nos navegadores com endereços brasileiros

Hola (*Brasil*)			
Navegador	*Firefox*	*Google Chrome*	*Edge*
IP	191.96.70.28	191.96.71.44	191.96.70.28
Localidade do *Proxy*			
Todos os testes	São Paulo	São Paulo	São Paulo
Whoer			
Proxy	Não.	Falha-	Não
Anonymizer	Sim	Falha	Sim
WebRTC	Sim	Não	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

. A página do *Whoer* indicou o *Firefox* e o *Edge* que estariam utilizando algum tipo de *Anonymizer* enquanto o *Google Chrome* falhou em renderizar essa parte da página. Todos os navegadores tiveram problemas menores sem exceção.

A capital do Afeganistão foi indicada como a origem dos endereços *IPs* pelo *IPlocation* como foi indicado na Tabela 21. Nos testes do *Whoer* teve resultados muitos semelhantes ao do teste anterior utilizando um *IP* brasileiro.

Tabela 21 - *Hola* configurado nos navegadores com endereços afegãos

Hola (*Afeganistão*)			
Navegador	*Firefox*	*Google Chrome*	*Edge*
IP	125.213.193.10	125.213.1210.18	149.54.3.179
Localidade do *Proxy*			
Todos os testes	*Kabul*	*Kabul*	*Kabul*
Whoer			
Proxy	Não	Falha	Não
Anonymizer	Sim	Falha	Sim
WebRTC	Sim	Não	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

3.2.4.2 CLIENTES PRÓPRIETÁRIOS

A segunda etapa dos testes com *VPNs* foi utilizando os clientes proprietários da *TunnelBear* e da *Windscribe*. O primeiro foi testado um endereço brasileiro e americano e no outro foi testado somente um endereço americano.

Um ponto interessante em relação aos clientes proprietários para com extensões de *VPNs* é a centralização do primeiro grupo que faz que todos os navegadores usem a mesma conexão.

Com o *Windscribe* na avaliação de localização foram mostradas cidades na região da costa oeste americana, porém com *Edge* falhando em gerar a página.

A Tabela 22 aponta que o teste da *Whoer*, todos os navegadores indicaram a utilização de um Anonymizer além de problemas secundários. Com *Firefox* também resultou problemas de *WebRTC* de alguma *LAN* que exibiu um vazamento de *IPs* sem vazar os dados da rede local.

Tabela 22 - *VPN Windscribe* cliente próprio no Windows com *IP* americano

***VPN* Windscribe(Brasil)**			
Navegadores	*Firefox*	*Edge*	*Google Chrome*
Localidade da VPN			
I2PLocation	*Atlanta*	Falha	*Atlanta*
ipinfo.io	*Cape Coral*	Falha	*Cape Coral*
DB-IP	*Washington*	Falha	*Washington*
Whoer			
Proxy	Não	Não	Não
Anonymizer	Sim	Sim	Sim
WebRTC	Sim	Não	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

Com o *TunnelBear* com endereço de *IP* brasileiro teve indicação em todos os testes de pertencer a cidade de São Paulo. Pelo teste da *Whoer* o campo *Anonymizer* apresentou um resultado positivo como é descrito na Tabela 23 nos três navegadores estando escrito *VPN*. Também apresentaram problemas menores. O *Firefox* novamente teve um vazamento *WebRTC* com valores que indicavam uma rede externa.

Tabela 23 - *VPN TunnelBear* cliente próprio no Windows com *IP* brasileiro

VPN TunnelBear (Brasil)			
Navegadores	*Firefox*	*Edge*	*Google Chrome*
Localidade do *Proxy*			
Todos os testes	São Paulo	São Paulo	São Paulo
Whoer			
Proxy	Não	Não	Não
Anonymizer	Sim	Sim	Sim
WebRTC	Sim	Não	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

Com o *TunnelBear* com um endereço americano o resultado apontado para *New York City* e dois para *Clifton* pela análise do *IPlocation*. No *Whoer*, o *Firefox* e o *Edge* sendo detectados e o *Google Chrome* com problemas para renderizar o topo do

teste demostrados na Tabela 24. Novamente, todos os navegadores tiveram problemas secundários e o *Firefox* sofrendo de *leak* de *WebRTC* de outra rede.

Tabela 24 - *VPN TunnelBear* cliente próprio no *Windows* com *IP* americana

VPN TunnelBear (*EUA*)			
Navegadores	*Firefox*	*Edge*	*Google Chrome*
Localidade da *VPN*			
I2PLocation	*New York City*	*New York City*	*New York City*
ipinfo.io	*Clifton*	*Clifton*	*Clifton*
DB-IP	*Clifton*	*Clifton*	*Clifton*
Whoer			
Proxy	Não	Não	Falha
Anonymizer	Sim	Sim	Falha
WebRTC	Sim	Não	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

3.2.4.3 *OPENVPN*

A primeira etapa para realizar o teste com *OpenVPN* foi seguir as instruções no *site* que oferecia o uma conexão gratuita para a utilização dela.

Na tabela 25 O teste no *IPlocation* resultou com endereço sendo americano para os três *browsers*.

Tabela 25 – *OpenVPN* – Endereço de *IP* dos EUA

OpenVPN			
Navegadores	*Firefox*	*Google Chrome*	*Edge*
Localidade da *VPN*			
I2PLocation	*Manassas*	*Manassas*	*Manassas*
ipinfo.io	*Washington D.C.*	*Washington D.C.*	*Washington D.C.*
DB-IP	*Las Vegas*	*Las Vegas*	*Las Vegas*
Whoer			
Proxy	*Maybe*	*Maybe*	*Maybe*
Anonymizer	Sim	Sim	Sim
WebRTC	Sim	Não	Não
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

No teste do *Whoer* alegou que as conexões talvez usassem *proxy* no campo de *proxy*, o *Anonymizer* apontou o uso de *VPN* além dos outros campos indicarem problemas secundários que foram detalhados também na Tabela 25. O *Firefox* também teve um vazamento *WebRTC* de uma *LAN* externa.

3.2.4.4 *VPN* DO OPERA

O navegador *Opera* tem uma *VPN* integrada ao mesmo. Para ativar a mesma basta habilitar ela nas opções avançadas e escolher o conjunto de servidores que deseja usar.

Na Tabela 26 teste com *IPlocation* falhou em gerar um resultado legível. No *Whoer* na opção *Anonymizer* indicou *PROXY* apesar da opção que marca o uso de *proxy* ter indicado negativo. Também houve problemas secundários e o uso de *DNS* Holandês.

Tabela 26 - Navegador *Opera* (*VPN* embarcada)

***Opera (VPN* nativa)**	
Whoer	
Proxy	Não
Anonymizer	*Sim*
WebRTC	Não
Problemas secundários	Sim

Fonte: Próprio autor

3.2.5 *I2P*

O primeiro passo para execução dos testes foi configurar o *Firefox* para utilizar o *I2P* e o seu *outproxy*. Na Tabela 27 assinala que no teste do *Whoer* houve um vazamento de *WebRTC*, problemas secundários e a utilização de *DNS* do Canada. Também na opção *Anonymizer* foi apresentado como utilizando o *Tor.*

Por algum motivo não foi possível utilizar a conexão do *I2P* para acessar o *site* do *JonDonym*.

Tabela 27 - *I2P* configurado no Firefox

Firefox (I2P)	
Whoer	
Proxy	Não
Anonymizer	*Tor*
WebRTC	Sim
Problemas secundários	Sim

Fonte: Próprio autor

A segunda etapa foi quando configurou o *I2P* para funcionar no gerenciador de *proxies* do sistema, sendo assim possível realizar os testes no *IPlocation*. Foi indicado as localidades de *Oslo*, *AL* e *Vikebukt*. Novamente a opção *Anonymizer* indicou como se *I2P* fosse o *Tor* e teve problemas secundários como indica a Tabela 28. No *Firefox* houve vazamento de *WebRTC* e no *Google Chrome* houve a utilização de *DNSs* Belgas e Holandeses.

Tabela 28 - I2P configurado no sistema

Configuração Manual de Proxy(I2P)		
Navegadores	*Firefox*	*Google Chrome*
Localidade do *outproxy*		
I2PLocation	*Oslo*	*Oslo*
ipinfo.io	*Al*	*Al*
DB-IP	*Vikebukt*	*Vikebukt*
Whoer		
Proxy	Não	Não
Anonymizer	*Tor*	*Tor*
WebRTC	Sim	Não
Problemas secundários	Sim	Sim

Fonte: Próprio autor

3.2.6 *TOR*

O primeiro teste utilizando o *Tor* foi através da configuração de rede do *Firefox*. O teste com o *IPlocation* resultou com endereço sendo *Schowalde* indicada uma vez e *Armstadam* duas. No teste do *Whoer* houve um vazamento de *WebRTC* da *LAN* local no *Firefox*, como indica a Tabela 29, e o *Anonymizer* indicou a utilização do *Tor*, houve utilização dos *DNS* da ISP local, além de problemas secundários.

Tabela 29 - *Tor* configurado no Firefox

Firefox (Tor)	
Localidade do *node* e saída	
I2PLocation	*Armstadam*
ipinfo.io	*Schowalde*
DB-IP	*Armstadam*
Whoer	
Proxy	Não
Anonymizer	*Tor*
WebRTC	Sim
Problemas secundários	Sim

Fonte: Próprio autor

O segundo método foi através da configuração direto no S.O. O teste com o *IPlocation* resultou com *Paris* sendo indicada 2 vezes e *Armstadam* uma vez. O endereço de *IP* do este é diferente do teste que foi configurado direto no *Firefox*. No caso do *Google Chrome* o teste falhou no primeiro campo, porém deu resultados similar ao Edge nos outros dois. No *Edge*, *Montreal* foi indicado as três vezes sendo a terceira com o complemento de *Villa*-Marie.

Na Tabela 30, o teste do *Whoer* foi indicada a utilização do *Tor* nos produtos da *Mozilla* e *Microsoft* e problemas secundários em todos. No caso do *Firefox* houve vazamento de *WebRTC* com dados da rede local.

Tabela 30 - *Tor* configurado no sistema

Configuração Manual de Proxy (Tor)			
Navegadores	*Firefox*	*Google Chrome*	*Edge*
Localidade do *node* e saída			
I2PLocation	Paris	Falha	Montreal
ipinfo.io	Armstadam	Montreal	Montreal
DB-IP	Paris	Montreal(Vila-Marie)	Montreal(Vila-Marie)
JonDonym			
Your IP	*Tor*	*Tor*	*Tor*
Whoer			
Proxy	Não	Falha	Não
Anonymizer	*Tor*	Falha	*Tor*
WebRTC	Sim	Não	Sim
Problemas secundários	Sim	Sim	Sim

Fonte: Próprio autor

O terceiro método foi através do navegador *Brave*, que o *Tor* embarcado nele da mesma forma que no *Tor Browser Bundle*. O teste com o *IPlocation* resultou com endereço sendo de *Amsterdam* e *New York City*, mas em cada aba ele ganhava um endereço diferente. No teste do *JonDonym* com o *Brave* mostrou que utilizava o *Tor*. No teste do *Whoer* a opção de *proxy* foi marcada como afirmativa, o *Anonymizer* indicou o uso do *Tor* como é descrito na Tabela 31.

Tabela 31 - *Brave* – aba anônima com *Tor* habilitado

Brave (Tor)	
Localidade do *node* e saída	
I2PLocation	Amsterdam
ipinfo.io	New York City
DB-IP	New York City
JonDonym	
Your IP	*Tor*
Whoer	
Proxy	Não
Anonymizer	*Tor*
WebRTC	Não
Problemas secundários	Não

Fonte: Próprio autor

O quarto método foi como o *Jondofox*. Ele apresentou com seu nó de saída sendo de Paris ou de outra localidade francesa. Como a Tabela 32 detalha, no *Whoer* ele foi indicado por usar o *Tor*, mas nada além disso.

Tabela 32 - *Tor JondoFox*

Jondofox (Tor)	
Localidade do *node* e saída	
I2PLocation	Paris
ipinfo.io	Paris
DB-IP	Clichy-sous-Bois
JonDonym	
Your IP	*Tor*
Whoer	
Proxy	Não
Anonymizer	*Tor*
WebRTC	Não
Problemas secundários	Não

Fonte: Próprio autor

O último método foi utilizando do *Tor Browser Bundle* com a configuração de segurança mediana. A Tabela 33 detalha seus resultados começando com o teste com o *IPlocation* resultou em 2 locais diferentes na Europa no teste do *Whoer* indicou a utilização de algum *Anonymizer*, mas não indicou explicitamente o uso do *Tor.* Já no teste do *JonDonym* resultou uma resposta positiva para o uso do *Tor.*

Tabela 33 - *Tor* Browser *Bundle* - configuração de segurança média

Tor Browser Bundle	
Localidade do *node* e saída	
I2PLocation	Amsterdam
ipinfo.io	Schowalde
DB-IP	Amsterdam
JonDonym	
Your IP	*Tor*
Whoer	
Proxy	Não
Anonymizer	Sim
WebRTC	Não
Problemas secundários	Sim

Fonte: Próprio autor

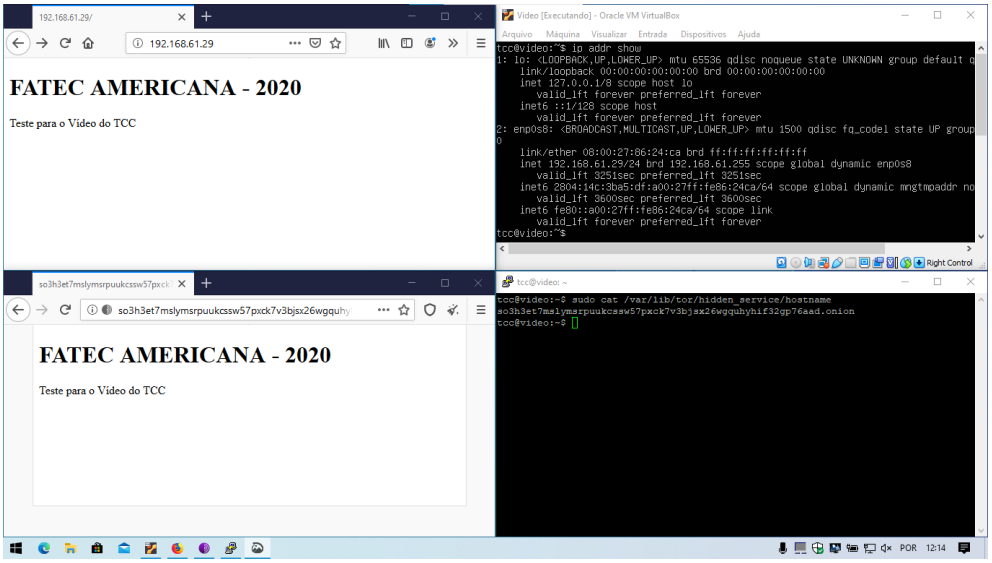
3.3 TESTE DE HIDDEN SERVICE

O teste foi realizado utilizando uma máquina virtual como o servidor *web* em no *host* hospedeiro e um computador separado. A virtualização foi realizada através do *VirtualBox* utilizando uma versão do *Ubuntu* feita especificamente para servidores.

O processo de testar a conexão com servidor era feito através do acesso do endereço de *IP* na *LAN* incialmente e após o sucesso da conexão era realizada através do *Tor Browser Bundle*.

Figura 10 mostra tanto o *Firefox* com o *Tor Browser Bundle*. acessando a página de teste, como a visualização do endereço de IP da máquina virtual e o programa *Putty* exibindo o endereço de *hidden service*.

Figura 10 – Teste do *Web* Service na *localhost*.



Fonte: Próprio autor

Na Figura 11 mostra parte do *log* de uma conexão entre um navegador do computador hospedeiro conectando ao servidor *web* na máquina virtual.

Figura 11 – Teste do *Web Service* na *localhost* no *Wireshark*

138 6.635783	2804:14c:3ba5:df:8c30:144:37cf:2beb	2804:14c:3bf9::c937:e9ca	TCP	75 60711 → 80 [ACK] Seq=1 Ack=1 Win=514 Len=1
139 6.647797	2804:14c:3bf9::c937:e9ca	2804:14c:3ba5:df:8c30:144:37cf:2beb	TCP	86 80 → 60711 [ACK] Seq=1 Ack=2 Win=242 Len=0 SLE=1 SRE=2
202 16.648493	2804:14c:3ba5:df:8c30:144:37cf:2beb	2804:14c:3bf9::c937:e9ca	TCP	75 [TCP Keep-Alive] 60711 → 80 [ACK] Seq=1 Ack=1 Win=514 Len=1
203 16.660190	2804:14c:3bf9::c937:e9ca	2804:14c:3ba5:df:8c30:144:37cf:2beb	TCP	86 [TCP Keep-Alive ACK] 80 → 60711 [ACK] Seq=1 Ack=2 Win=242 Len=0 SLE=1…
226 21.957253	2804:14c:3ba5:df:8c30:144:37cf:2beb	2804:14c:3bf9::c937:e9ca	TCP	74 60711 → 80 [FIN, ACK] Seq=2 Ack=1 Win=514 Len=0
228 21.970241	2804:14c:3bf9::c937:e9ca	2804:14c:3ba5:df:8c30:144:37cf:2beb	TCP	74 80 → 60711 [FIN, ACK] Seq=1 Ack=3 Win=242 Len=0
230 21.970366	2804:14c:3ba5:df:8c30:144:37cf:2beb	2804:14c:3bf9::c937:e9ca	TCP	74 60711 → 80 [ACK] Seq=3 Ack=2 Win=514 Len=0

Fonte: Próprio autor

Nesse outro exemplo, na Figura 12, mostra uma parte da conexão entre um navegador fora do hospedeiro solicitando a página *Web*.

Figura 12 – Teste do *Web Service* na *localhost* por cliente externo na *LAN*

6 1.587610	192.168.61.24	192.168.61.29	TCP	74 60380 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=170…
7 1.588204	192.168.61.29	192.168.61.24	TCP	74 80 → 60380 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=…
8 1.588459	192.168.61.24	192.168.61.29	TCP	66 60380 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1708002652 TSecr=2…
9 1.588897	192.168.61.24	192.168.61.29	HTTP	621 GET / HTTP/1.1
10 1.589364	192.168.61.29	192.168.61.24	TCP	66 80 → 60380 [ACK] Seq=1 Ack=556 Win=64640 Len=0 TSval=2708298197 TSecr…
11 1.591243	192.168.61.29	192.168.61.24	HTTP	592 HTTP/1.1 200 OK (text/html)
12 1.591712	192.168.61.24	192.168.61.29	TCP	66 60380 → 80 [ACK] Seq=556 Ack=527 Win=64128 Len=0 TSval=1708002655 TSe…
19 6.596421	192.168.61.29	192.168.61.24	TCP	66 80 → 60380 [FIN, ACK] Seq=527 Ack=556 Win=64640 Len=0 TSval=270830320…
20 6.638288	192.168.61.24	192.168.61.29	TCP	66 60380 → 80 [ACK] Seq=556 Ack=528 Win=64128 Len=0 TSval=1708007702 TSe…

Fonte: Próprio autor

Na Figura 13 a comunicação entre o *Tor Browser Bundle* agindo como cliente e a *web* server na máquina virtual. Na imagem mostra o *IP* de *LAN* da máquina hospedeira se conectando na rede *Tor* e o *IP* do servidor recebendo a transmissão de dados.

Figura 13 – *Tor Browser* na máquina hospedeira acessando o hidden service

No.	Time	Source	Destination	Protocol	Length	Info
3	0.010064	46.4.78.148	192.168.61.29	TLSv1.2	602	Application Data
4	0.010583	192.168.61.29	46.4.78.148	TCP	66	52046 → 9001 [ACK] Seq=1 Ack=537 Win=3093 Len=0 TSval=36970974…
6	0.805940	192.168.61.28	147.135.4.38	TLSv1.2	590	Application Data
15	0.962369	147.135.4.38	192.168.61.28	TCP	60	9001 → 52575 [ACK] Seq=1 Ack=537 Win=352 Len=0
16	0.964791	147.135.4.38	192.168.61.28	TLSv1.2	590	Application Data
17	0.966858	192.168.61.28	147.135.4.38	TLSv1.2	590	Application Data

Fonte: Próprio autor

Por fim, na Figura 14 mostra-se uma conexão entre um computador independente e o servidor *Web*. A conexão pareceu demorar tanto quanto na máquina local.

Figura 14 – Cliente externo acessando o hidden service

9	3.924862	46.4.78.148	192.168.61.29	TLSv1.2	602 Application Data
10	3.925432	192.168.61.29	46.4.78.148	TCP	66 52046 → 9001 [ACK] Seq=1 Ack=537 Win=3093 Len=0 TSval=36981010…
11	3.927058	192.168.61.29	46.4.78.148	TLSv1.2	602 Application Data
12	4.159413	46.4.78.148	192.168.61.29	TCP	66 9001 → 52046 [ACK] Seq=537 Ack=537 Win=2474 Len=0 TSval=389388…
13	4.159861	192.168.61.29	46.4.78.148	TLSv1.2	602 Application Data

Fonte: Próprio autor

3.4 TESTES AVANÇADOS

3.4.1 *PROXY* BRASILEIRO

O primeiro teste avançado foi realizado utilizando um *IP* brasileiro de nível de anonimato classificado como *elite* tendo endereço como 186.193.251.10. A porta desse *proxy* era 80, mas conseguiu trabalhar com uma conexão criptografada.

A escolha de realizar um teste com *IP* especificamente brasileiro se deve ao fato que algumas das vulnerabilidades de privacidade e anonimato são analisadas através de comparações de idiomas e horários entre idiomas e horários dos aplicativos, sistema e o número do *IP* válido.

Isso significa que caso o usuário estiver utilizando um sistema operacional ou navegador com idioma ou horário diferente do que deveria ser de acordo com a região do IP, ele recebera um resultado do teste *Whoer* que poderá ser considerado um falso positivo.

Antes da configuração no teste do *JonDonym* com o *Firefox* não indicou a utilização de *proxy* ou do *Tor*. No teste do *Whoer* aconteceu vazamento de *WebRTC* e indicou a utilização de portas comuns de *proxies*.

Posteriormente foi utilizado a extensão *µblock* e de sua opção para proteger contra vazamento de *WebRTC*. Porém, devido a situação de utilizar portas comuns, não foi possível esconder essa fraqueza.

3.4.2 *PROXY* ESTRANGEIRO

Foi escolhido para o teste de *proxy Elite* uma opção canadense com endereço de 198.27.67.35. A porta desse *proxy* era 3128.

Inicialmente, o teste do *JonDonym* com o *Firefox* não indicou a utilização de *proxy* ou do *Tor*. No teste do *Whoer* aconteceu vazamento de *WebRTC* com endereço da rede local e indicou a utilização de portas comuns de *proxies*. Também indicou incoerência entre endereço de *IP*, o idioma no sistema e/ou navegador, o horário e o *IP*.

Após a ativação do *µblock* e de sua opção para proteger contra vazamento de *WebRTC* e foi configurado o idioma do navegador e do sistema operacional e o horário para localidade do IP para mitigar esses problemas.

Porém, novamente devido a situação de utilizar portas comuns, não foi possível esconder o fato que era *proxy* sendo utilizado.

3.4.3 *VPN* ESTRANGEIRA

No teste sem modificações no *JonDonym* com o *Firefox* não indicou a utilização da *VPN* ou do *Tor*. No teste do *Whoer* aconteceu vazamento de *WebRTC*. Também indicou incoerência entre endereço de *IP*, o idioma no sistema e/ou navegador, o horário e o *IP*, diferença entre países do endereço dos *DNS* usado e o *IP* da conexão e indicou a conexão tinha *IP* suspeito.

Após a ativação do *µblock* e de marca opção para proteger contra vazamento de *WebRTC* foi configurado idioma do navegador e do sistema operacional e o horário para de acordo com a localidade do *IP*. Também foi necessário modificar o arquivo de configuração da *VPN* para utilizar um *DNS* americana como o do endereço *IP* utilizado. Porém, mesmo obtendo o resultado máximo através de configurar todos os requisitos de acordo com recomendações do *Whoer*, o endereço *IP* continuava indicando como suspeito.

4 CONSIDERAÇÕES FINAIS

Como foi observado nos testes, as redes sobrepostas apresentam um resultado misto. Devido as conexões do *Tor* não serem esteganográficas, elas vão servir principalmente como um túnel seguro de alta disponibilidade, se bem configurado para a *Internet*. Entretanto qualquer servidor poderá perceber que a conexão está vindo da

rede do *Tor*, caso a mesma não for ocultada por outro recurso, e poderá tomar medidas que atendam sua necessidade em relação a filtragem das conexões com essa origem.

Já alguns serviços de *proxies* e *VPNs* não são identificados como tais dando a possiblidade de serem classificados como uma conexão legitima em caso de uma tentativa de filtragem de conexões que procuram ocultar o endereço *IP* verdadeiro. Essas brechas na filtragem podem serem utilizadas para atividades inofensivas, como escolhera uma região diferente na *Netflix*, ou para o mais variado tipo de atividades do crime organizado cibernético.

Entretanto, esses serviços têm uma centralização ainda maior do que o *Tor*, o que leva uma fragilidade em caso de um ataque *hacker* contra sua infraestrutura, o que pode causar um vazamento em relação a informação dos seus clientes. Outro ponto negativo é que a maioria desses recursos são configurados por padrão para serem intermediário direto entre o requisitante e o servidor, enquanto o *Tor* tem por padrão três pulos entre as pontas.

Isso significa que é necessário entender esses softwares como ferramentas e não como truques de mágica ou soluções prontas de segurança e privacidade como foi divulgado no passado, além de analisar suas características para definir em quais cenários eles são mais eficazes para modelos variados de uso da *Internet*. Um excelente exemplo disso são as diferenças entre *VPNs* e *proxies*.

Em ambos parecem executarem as mesmas atividades à primeira vista, porém possuem diferenças como o uso comum de clientes proprietários e a capacidade de utilizar conexões *UTP* pelas *VPNs*. Também tem a questão que os *websites* que fazem a listagem dos *proxies* gratuitos normalmente trabalham de forma obscura para obter os endereços que oferecem.

Outro fato que deve ser lembrado é que dos serviços gratuitos, o melhor na maior parte dos casos de navegação na *Web* é *Tor Browser Bundle*. Mesmo o protocolo do *Tor* ser flexível para sua utilização na maioria dos programas que permitem a configuração da conexão de rede, em caso de uso direto em um navegador, a melhor opção é utilizar através do *Tor Browser Bundle* o qual já vem com um *daemon* embarcado e o navegador *Firefox ESR* customizado para uma

melhor utilização do *Tor*, evitando problemas que talvez aconteceriam quando um usuário desavisado configurasse diretamente o *Tor* para funcionar em seu navegador.

Com foco no aumento da usabilidade, o *I2P* começou seguir o mesmo caminho embrancando seu *software* com uma versão *ESR* do *Firefox* customizada. No caso dessa experiência tiver sucesso, isso propiciara uma maior robustez devido o modelo de funcionamento da rede ser *P2P*, sendo com ampliação do número de usuários pode causar um aumento na disponibilidade, privacidade e velocidade da conexão.

Outro ponto interessante que o *Tor* se diferencia dos demais é a questão da influência e do dinheiro. O *Tor* é de longe o serviço de rede sobreposta público que pode ser taxado de *Darknet* que mais recebe doações financeiras e atenção pela academia, causando um melhoramento muito mais ágil se comparado com outros serviços.

Por causa disso um *hidden service* trabalhe com o *Tor* recebera mais atenção do que em qualquer outra rede, significando que um recurso no segundo colocado, *I2P*, terá menos visibilidade. Também existe a questão em aberto de que o *I2P* pode ter sido ultrapassado pelo *Tor* na disputa de criar uma *Internet* dentro da *Internet* mesmo sendo seu objetivo primário enquanto para outro projeto é uma questão secundária.

Ambos os programas citados no parágrafo anterior são mais seguros juridicamente ao seu usuário do que serviços pagos ou gratuitos de *VPNs* ou *proxies* por causa de diversos fatores, mas especialmente por causa de sua descentralização jurídica.

Um ponto importante que deve ser sempre relembrado é que sempre será mais custoso para um atacante quebrar a criptografia e as defesas de um serviço que protege as identidades de seus usuários em tempo real do que é para defender a mesma. Como resultado as agências de inteligências ao redor do mundo necessitam de um orçamento grande quando resolvem atacar esse tipo de programas, além de também necessitar uma equipe técnica especializada para procurar falhas na configuração ou na lógica matemática dos algoritmos de criptografia.

Em relação a questão semântica de *Deep Web*, *Dark Web* e *Darknet* pode-se dividir da seguinte forma:

- *Deep Web* para páginas *Web* que não estão indexadas nos servidores de busca, o que faria *Dark web* ser parte dela
- *Darknet* para serviços de uma forma geral que usam redes sobrepostas para serem uma *Internet* separada da *Internet*
- *Dark Web* seria páginas *webs* que necessita a utilização uma rede sobreposta para ser acessada
- Em caso de motores de busca serem capazes de indexar serviços separados de servidores *Webs,* poderia também ter um *DeepNet* onde seriam servidores de serviços que não são de páginas que não foram indexados.

Entretanto, essas definições devem servir de forma mais introdutória e com evolução do entendimento das ferramentas usadas, deve começar chamar o *Tor* e o *I2P* pelo o que eles realmente são, redes sobrepostas. Da mesma forma que redes corporativas podem ter domínios não padronizados, o *Tor* e *I2P* são como um conjunto de *intranets* públicas dentro da *Internet*.

Esse trabalho começou com uma palavra da moda dos anos da década passada, a *Deep Web*, e termina com uma palavra que pode cair na moda, a *splinternet*, que nada mais é que uma versão repaginada para o conceito do que era chamado de soberania virtual ou balcanização da *Internet*, tendo como seu maior exemplo o Grande *Firewall* da China.

Isso significa que governos nacionais tentaram intervir cada vez mais na *Internet* para resguardar seus interesses da mesma forma que empresas privadas mudam seus termos de serviços. Isso pode apresentar pontos positivos e pontos negativos para maior parte dos grupos sociais, porém indivíduos específicos podem serem atingidos caso o mundo caminhar nessa direção sendo que os efeitos dessa situação poderiam ser amenizados pelo uso do *Tor*.

Nesse cenário, o *Tor* serviria para maior parte do mundo para o que serve para pessoas em países em que a Internet não é tão livre, sendo uma ferramenta gratuita e poderosa para esquivar da censura governamental. Já o caso do *I2P*, ele precisa se aperfeiçoar para se tonar *Intranet* pública da *Internet* como eles almejam. O que nessa década teve o ar exótico, pode acabar se tornado na próxima década o que *Naspter*

e os programas *P2P* foram para compartilhamento de arquivos para os anos 2000 tendo seu uso algo como corriqueiro.

REFERÊNCIAS BIBLIOGRÁFICAS

ANNESSI, R.. **Lower-latency anonymity** latency reduction in the *Tor* network usingcircuit-level round-trip-time measurements. 2014. Disponível em: < http s://pdfs.semanticscholar.org/33ab/9cf0f5b1e65e953e26e32f6aee165760793c.pdf>. Acesso em: 23 out. 2019.

ANTICOUNTERFEITING COMMITTEE. **Anticounterfeiting on the *dark* web.** 2015. Disponível em: < https://www.inta.org/Advocacy/Documents/2015/ ACC%20Dark%20Web%20 Report.pdf >. Acesso em: 22 set. 2019

BALASUBRAMANIAN, N. **Survey paper on rising threats of subverting privacy infrastructure.** Disponível em: < https://arxiv.org/pdf/1612.05806.pdf>. Acesso em: 17 novembro. 2019

BERNERS-LEE, T. **Universal resource identifiers in www**. CERN, 1994. Disponível em: <https://tools.ietf.org/html/rfc1630>. Acesso em: 22 set. 2019

BIDDLE, P., et at . **The *darknet* and the future of content distribution.** 2003. Disponível em: <https://pdfs.semanticscholar.org/8dc3/ edd5dd0a51412781fd2af22e58da8ae9edac.pdf >. Acesso em: 22 set. 2019.

BRINKLIN, D. . **Friend-to-friend networks.** 2000. Disponível em: < http://www.bricklin.com/f2f.htm>. Acesso em: 22 set. 2019.

BLOWERS, M. *(*ed) . **Evolution of cyber technologies and operations to 2035**.. Switzerland: Springer, 2015.

BRANDON. **New feature**: ssl for users. Disponível em: < *https*://www.dal .net/news/shownews.php?id=67>. Acesso em: 20 nov. 2019.

CIANCAGLINI, V., et al. **The surface:**exploring the *deep* web**..** 2015. Disponível em: < https://documents.trendmicro.com/assets/wp/wp_below_the_surface.pdf >. Acesso em: 22 set. 2019.

CROSS, S. ***Deep* web -** olhar digital 2019**.** 2013**.** Disponível em: < https://www.youtube.com/watch?v=gZ3kzI4tLkw>. Acesso em: 22 set. 2019.

DINGLEDINE,R.. **Pre-alpha: run an onion *proxy* now!**. Disponível em: < https://archives.seul.org/or/dev/Sep-2002/msg00019.html>. Acesso em: 16 out. 2019

DINGLEDINE, R.; MATHEWSON, N.;SYVERSON, P. ***Tor*:** the second-generation onion router**.** 2004. Disponível em: < https://svn-archive.torproject.org /svn/projects/design-paper/tor-design.pdf >. Acesso em: 22 set. 2019.

FREE-PROXY.CZ. **Frequently asked questions** Disponível em: < http://free-proxy.cz/en/faq> Acesso em: 17 nov. 2019.

FIELDING, R. ET AL **Hypertext transfer protocol --** *HTTP* /1.1. W3C/MIT, 1999. Disponível em: <https://tools.ietf.org/html/rfc2616>. Acesso em: 08 out. 2019

HOLDEN, E.. **An introduction to *Tor* vs *I2P***. Disponível em: < https://archives.seul.org/or/dev/Sep-2002/msg00019.html>. Acesso em: 05 nov. 2019

HOPPER, N.; VASSERMAN E. Y.; CHAN-TIN ERIC . **How much anonymity does networklatency leak?.** University of Minnesota. 2010. Disponível em: < http://www.scielo.br/pdf/tinf/v25n1/a04v25n1.pdf>. Acesso em: 16 out. 2019.

GNUNET E.V . **Gnunet reference manual**. Disponível em: < https://docs.gnunet.org/handbook/gnunet.html#Introduction>. Acesso em: 16 out. 2019.

I2P TEAM. **A gentle introduction to how *I2P* works**. Disponível em: <https://getI2P.net/pt-br/docs/how/intro>. Acesso em: 16 out. 2019.

ICANN. **Guia do iniciante para participação na *icann*.** Disponível em: < https://www.icann.org/en/system/files/files/participating-08nov13-pt.pdf>. Acesso em: 16 out. 2019

IHLENFELD, J. **Anonyme p2p-software freenet 0.7 soll nutzer besser schützen**. 2018. Disponível em: < https://www.golem.de/0805/59592.html>. Acesso em: 16 out. 2019

INETDAEMON. **DNS hierarchy**. 2018. Disponível em: < https://www.inetdaemon.com/tutorials/internet/dns/operation/hierarchy.shtml>. Acesso em: 22 set. 2019.

KARRENBERG, D. **The *internet* domain name system explained for non-experts.** 2004. Disponível em: < https://www.internetsociety.org/wp-content/uploads/2017/09/The-Internet-Domain-Name-System-Explained-for-Non-Experts-ENGLISH.pdf>. Acesso em: 22 set. 2019.

KLENSIN, J. **Role of the domain name system (DNS)**. 1993. Disponível em: < https://tools.ietf.org/html/rfc3467>. Acesso em: 22 set. 2019

LE BLOND, S. . **Towards efficient traffic-analysis resistantanonymity networks.** 2000. Disponível em: < https://www.freehaven.net/anonbib/papers/sigcomm13-aqua.pdf>. Acesso em: 22 set. 2019.

MACHADO, A. .**Há 40 anos, surgia a Arpanet,** o embrião da *Internet*. Disponível em: <https://www.nic.br/noticia/na-midia/ha-40-anos-surgia-a-arpanet-o-embriao-da-internet/ >. Acesso em: 15 set. 2019.

METZ, C. **Why does the net still work on christmas?** paul mockapetris. 2012. Disponível em: < https://www.internethalloffame.org/blog/2012/07/23/why-does-net-still-work-christmas-paul-mockapetris>. Acesso em: 22 set. 2019 16:16.

MITCHELL, B. **9 best free *anonymous proxy* servers.** Lifewire

Disponível em: < https://www.lifewire.com/free-anonymous-web-proxy-servers-818058> 18 de ago. 2019

MONTEIRO, S. D.; FIDENCIO, M. V. . **As dobras semióticas do ciberespaço:** da *web* visível à invisível**.** 2013. Disponível em: <http://www.scielo.br/pdf/tinf/v25n1/a04v25n1.pdf>. Acesso em: 22 set. 2019.

MORAIS, C.T.Q.; LIMA, J.V ; FRANCO, S.R . **Conceitos sobre *internet* e web** . Porto Alegre, RS: Edi*Tor*a DA UFRGS, 2012. Disponível em: <http://www.ufrgs.br/sead/servicos-ead/publicacoes-1/pdf/Conceitos_Internet_e_Web.pdf>. Acesso em: 11 abr. 2008.

NIC. **Defense data network newsletter**. Disponível em: < https://www.rfc-editor.org/rfc/museum/ddn-news/ddn-news.n26.1>. Acesso em: 15 set. 2019.

NORSAD. **Arpanet** Disponível em: < https://www.norsar.no/about-us/hisTory/arpanet-article774-270.html>. Acesso em: 15 set. 2019.

NOVA PROXY. ***Proxy* anonymity levels** Disponível em: < https://www .proxynova.com/proxy-server-list/>. Acesso em: 17 nov. 2019.

NOVA PROXY. **Free *proxy* list -** list of open *proxy* servers Disponível em: < https://www.proxynova.com/proxy-articles/proxy-anonymity-levels-explained/>. Acesso em: 17 nov. 2019.

RESCORLA, E., T. **Http over tls**. RTFM, Inc, 2000. Disponível em: < https://tools.ietf.org/html/rfc2818>. Acesso em: 08 out. 2019

SHABBAR, K.; ZINEIR-HEYWOOD, A. N. . **Effects of Shared Bandwidth on Anonymity of the I2P Network Users**. Disponível em: < https://sci-hub.se/10.1109/spw.2017.19> Acesso em: 27 nov. 2019

SHABBAR, K.; ZINEIR-HEYWOOD, A. N. . **Weighted factores for measuring Anonymity Services:** a case study on tor, jondonym, and i2p. Disponível em: < https://pdfs.semanticscholar.org/6070/c8e4aa2242082228b35acf1c49eb3cb24c40.pdf> Acesso em: 27 nov. 2019

SPELTA, L.; SOARES, H. *(*ed*).* **Cartilha de acessibilidade na web.** São Paulo, SP; DB Comunicação Ltda, [n.d.]. Disponível em: < https://www.w3c.br/pub/Materiais/PublicacoesW3C/cartilha-w3cbr-acessibilidade-web-fasciculo-I.html> Acesso em: 18 set. 2019

STEWART, B. ***ARPANET –*** The First *Internet*. Disponível em: < https://www.livinginternet.com/i/ii_nsfnet.htm >. Acesso em: 19 nov. 2019

STEWART, B. **NSFNET -** National Science Foundation Network. Disponível em: < https://www.livinginternet.com/i/ii_nsfnet.htm >. Acesso em: 16 out. 2019

THE ELECTRONIC FRONTIER FOUNDATION. **How *HTTP S* and *Tor* work together to protect your anonymity and privacy** Disponível em: < https://www.eff.org/ pages/tor-and-https>. Acesso em: 22 set. 2019.

THE FREENET PROJECT. **About**. Disponível em: < https://freenetproject.org/pages/about.html>. Acesso em: 16 out. 2019.

THE *TOR* PROJECT, INC. ***Tor***: overview Disponível em: < *https*://2019.ww w.torproject.org/about/overview>. Acesso em: 05 nov. 2019.

THE *TOR* PROJECT, INC. ***Tor***: onion service protocol Disponível em: < *https*://2019.www.torproject.org/docs/onion-services.html.en>. Acesso em: 05 nov. 2019

WEINBERGER, S. . **what is siprnet?** Disponível em: <https://www.nic.br/noticia/na-midia/ha-40-anos-surgia-a-arpanet-o-embriao-da-internet/ >. Acesso em: 15 set. 2019.

WOOD, J. . **The *darknet*:** a digital copyright revolution**.** 2010. Disponível em: < http://jolt.richmond.edu/jolt-archive/v16i4/article14.pdf>. Acesso em: 22 set. 2019.

WU, T. **Impérios da comunicação:** do telefone à *internet*, da at&t ao google. Tradução de Claudio Carina. Rio de Janeiro: Zahar, 2012.