Universidade do Minho
Escola de Engenharia

Carlos Eduardo Ribeiro Machado

Navegação Segura - Análise do Uso de HTTPS na Perspectiva do Utilizador Final

julho de 2020

Navegação Segura - Análise do Uso de HTTPS na Perspectiva do Utilizador Final
Carlos Eduardo Ribeiro Machado
UMinho | 2020

Direitos do Autor e Condições de Utilização do Trabalho Por Terceiros

Abstract

The Internet emerged in the late sixties in a scenario marked by the race of world hegemony between USA and USSR. Besides military applications, it was also initially used by researchers, academics, and college students, enabling file transfer between hosts. After the nineties the Internet reached the general public. It was then focused on other purposes, such as access to hypermedia, social networks, advertising and even products sale.

Given the diversification of these accesses, the adoption of protocols for safe browsing has become essential to protect user's information. Combined with the classification of encrypted traffic, using appropriate techniques for this purpose, this paper aims to analyze the use of HTTPS protocol in various browsing scenarios once considered safe. Through testing scenarios, this research intends to verify changes and impacts that this protocol promotes regarding the data collection from the users during the Internet access experience.

Keywords: HTTPS, User, Web security.

Resumo

A Internet surgiu no final da década de sessenta em um cenário marcado pela disputa da hegemonia mundial entre EUA e URSS. Além de aplicações militares, ela foi utilizada inicialmente por pesquisadores, académicos e estudante universitários, possibilitando a transferência de arquivos entre hospedeiros. A partir da década de noventa a Internet chegou ao grande público. Passou, então, a ser utilizada para outros propósitos, como o acesso a hipermídias, redes sociais, publicidade e até venda de produtos.

Diante da diversificação desses acessos, a adoção de protocolos para navegação segura tornou-se essencial para proteção das informações dos utilizadores. Aliado à classificação de tráfego encriptado, utilizando técnicas apropriadas para o efeito, este trabalho tem por objetivo analisar o uso do protocolo HTTPS em vários cenários de navegação considerados seguros. Através de cenários de teste, pretende-se verificar mudanças e impactos que este protocolo repercute quanto à exposição de dados na experiência de acesso à Internet de um utilizador final.

Palavras-Chave: HTTPS, Utilizador, Segurança na Web.

Agradecimentos

Presto aqui meus agradecimentos a todos que torceram por mim e aos que de alguma forma se fizeram fundamentais para a conclusão dessa etapa de minha vida. Em especial:

À minha família, pelo apoio incondicional e amor. Principalmente à minha mãe e minha irmã, por sempre acreditarem em mim, e ao meu pai, pois sei que está sempre ao meu lado.

À minha esposa Maíra, pelo suporte e por todo amor resistente a um oceano de distância, e toda sua família.

Aos meus tios José Ribeiro e Edvaldo, pelo amparo nos momentos de maior necessidade.

À minha orientadora, Professora Solange Rito Lima, e ao meu co-orientador, Professor Paulo Martins, expresso minha profunda gratidão pelo direcionamento, disponibilidade, apoio e pela oportunidade de aprender todos os dias durante nossa convivência.

A todos os Professores do MERSTEL, pela oportunidade de adquirir novos conhecimentos e trocar experiências.

À Universidade do Minho, por me aceitar como aluno e me permitir viver essa experiência incrível.

À Luz e ao Mário, pelo acolhimento, apoio e carinho.

Ao meus amigos Leandro, Rui, Edgar e Miguel, pela parceria e aprendizado nesses dois anos de mestrado.

Aos meus amigos Ivano, Pedro, Bruno, Adilson e Anderson, por estarem presentes em todos os momentos.

Declaração de Integridade

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ACM	Association for Computing Machinery Digital Library
ARPA	Advanced Research Projects Agency
ARPANET	Advanced Research Projects Agency Network
ASCII	American Standard Code for Information Interchange
CA	Certification Authority
CAIDA	Center for Applied Internet Data Analysis
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CERN	Conseil Européen pour la Recherche Nucléaire
CORE	Computing Research and Education Association of Australasia
DNS	Domain Name System
DOD	Departament of Defense
DoS	Denial of Service
DPI	Deep Packet Inspection
FTP	File Transfer Protocol
HRU	Harrison, Ruzzo, Ullman mode
HTML	Hypertext Makeup Languge
HTTP	Hypertext Terminal Protocol
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
IDE	Integrated Development Environment

IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMP	Interface Message Processors
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISP	Internet Service Providers
kbps	kilobits por segundo
LAN	LocalArea Network
LPI	Lightweight Packet Inspection
MAC	Message Authentication Code
MILNET	Military Network
MIT	Massachusetts Institute of Technology
MPLS	Multiprotocol Label Switching
NCSA	National Center for Supercomputing Applications
NPC	Network Control Protocol
NSA	Nacional Security Agency
NSF	National Science Foundation
NSFNET	National Science Foundation Network
PPPoE	Point–to–Point Protocol over Ethernet
QUIC	Quick UDP Internet Connections
RFC	Request for Comments
RGPD	Regulamentação Geral da Proteção de Dados
RMON	Remote Network Monitoring
RTT	Round Trip Time
SLR	Systematic Literature Review
SMNP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol

SNMP	SimpleNetwork Management Protocol
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network
WAN	Wide Area Network
WWW	World Wide Web

Capítulo 1

Introdução

1.1 Introdução

O sucesso da Internet transformou o HTTP (*Hypertext Terminal Protocol*) no principal protocolo de troca de informações por quase duas décadas. No início de sua utilização, não havia preocupação com a encriptação das informações entre clientes e servidores. Com o auxílio de ferramentas adequadas, os dados trocados poderiam ser facilmente interceptados por qualquer pessoa.

Devido à falta de privacidade das informações, ficou evidente a fragilidade da segurança na troca de dados e vulnerabilidade aos mais diversos tipos de ataques cibernéticos. Sendo assim, implementou-se o HTTPS (*Hypertext Transfer Protocol Secure*) que é o protocolo HTTP configurado sobre uma camada de segurança que aplica os protocolos SSL (*Secure Sockets Layer*) e TLS (*Transport Layer Security*), que por sua vez, proporcionam a transmissão de dados através de uma conexão encriptada. Esse protocolo permitiu a verificação da integridade do servidor e do cliente por meio de certificados digitais.

Nesse contexto, a classificação de tráfego teve um papel importante na análise de informação e passou a ter grande destaque, pois fatores como acréscimo de dispositivos de acesso à Internet e aumento das bases de dados passaram a ser alvo de grande interesse para indivíduos e organizações. Pelo fato de que redes com grandes dimensões necessitam lidar com grandes quantidades de tráfego de dados todos os dias, a classificação do tráfego possibilitou realizar melhorias de recursos, identificar anomalias de segurança e melhorar a qualidade de serviço. No entanto, a encriptação trouxe grandes desafios no processo de classificação do tráfego, o qual precisou se adaptar para continuar a obter informações significativas nos dados coletados.

Todas essas modificações tiveram impactos na experiência do utilizador final. Este trabalho procura identificar o impacto que a implantação do protocolo HTTPS causou quanto à vulnerabilidade e exposição de dados durante a navegação na Internet.

1.2 Motivação e Objetivos

A experiência de acesso à Internet modificou-se muito com o passar dos anos, em que a utilização de uma forma de navegação que preserve os dados dos utilizadores passou a ser obrigatória. Através do relatório de transparência do Google de julho/2019 [1] e do estudo de tendências da criptografia global da empresa Ncipher [2] tem-se evidências do crescimento da procura por acessos através de meios mais seguros. Mesmo assim, ainda há um grande volume de conteúdos disponibilizados na Internet que não ofereça nenhum tipo de segurança ao utilizador durante o acesso e o controle sobre esse aspecto ainda parece distante. Diante dessas informações, não fica explícito o comportamento de acesso e qual o verdadeiro impacto na experiência de navegação do utilizador. O estudo desse comportamento virtual torna-se relevante, visto que é uma forma de identificar o estágio atual de encriptação dos dados e como ele atua no acesso padrão de um utilizador final dependendo do meio de acesso escolhido.

Este trabalho procura identificar o impacto que a implantação do protocolo HTTPS causou quanto a vulnerabilidade e exposição dos dados durante a navegação da Internet. Alguns objetivos intermédios definidos como forma de atingir o objetivo principal deste trabalho incluem:

- analisar quais os impactos mais significativos do protocolo HTTPS na conexão do utilizador;
- verificar se os sites de busca realizam redirecionamento dos utilizadores de forma efetiva para endereços seguros;
- identificar e comparar comportamentos no acesso realizado pelo utilizador a partir de um computador pessoal com sistemas operacionais diferentes.

1.3 Estrutura da Dissertação

O presente Capítulo apresentou alguns pontos importantes sobre o processo evolutivo da Internet, em termos tecnológicos e de segurança, de forma breve e para contextualização inicial dos objetivos deste trabalho.

No Capítulo 2 destacam-se conceitos fundamentais base para o desenvolvimento deste de

trabalho. Além de uma visão geral da evolução da Internet em termos de tecnologias e segurança, protocolos de comunicação e princípios sobre classificação e análise do tráfego de rede.

No Capítulo 3 descreve-se de forma breve o processo de recolha de material referencial por intermédio de uma revisão sistemática da literatura. Como fruto dessa pesquisa, alguns trabalhos também são mencionados para ilustrar análises desenvolvidas na temática dessa dissertação.

No Capítulo 4 discute-se a metodologia implementada para criar um ambiente de testes e realizar a análise e o tratamento dos dados capturados. São descritos os processos de implementação do ambiente de teste em ambientes virtualizados, sistemas de testes e ferramentas para análise dos dados.

No Capítulo 5 aborda-se toda a análise dos dados coletados no processo de medição realizado no ambiente de experimentação. A investigação busca apresentar a existência de diferenças de acesso em dois sistemas operacionais, com três tipos de navegadores, endereços eletrónicos e vários motores de busca.

Finalmente, no Capítulo 6 apresentam-se as conclusões deste trabalho de dissertação com uma avaliação e reflexão de acordo com os resultados obtidos e objetivos inicialmente definidos. Também serão abordados possíveis ramificações para trabalhos futuros associados à temática desenvolvida.

Capítulo 2

Estado da Arte

Este capítulo tem como principal objetivo enquadrar os conceitos fundamentais deste trabalho, que são a origem, desenvolvimento, arquitetura e segurança da Internet. Também destaca os principais protocolos empregues para comunicação e o processo de evolução da segurança dos dados, além de apresentar os princípios essenciais sobre classificação e análise do tráfego de rede.

2.1 Internet

Dois ou mais computadores interligados entre si, de forma a compartilharem recursos físicos ou fazerem trocas de informações, definem uma rede de computadores. Também conhecida como a rede mundial de computadores, a Internet é a junção de um vasto conjunto de redes diferentes, a qual interliga milhões de dispositivos espalhados pelo mundo [3].

Esses dispositivos podem ser computadores pessoais, estações de trabalho, dispositivos móveis ou servidores, que armazenam e transmitem os mais diversos tipos de informações, como entretenimento (jogos, vídeos, música), troca de mensagens através de correio eletrónico (*e-mail*) e redes sociais em geral, comércio, grupos de discussão e conversação em tempo real.

A Internet passou a ser um novo espaço de convivência e interação entre as pessoas, sem nenhum tipo de fronteira e descentralizado. Segundo o filósofo Pierre Lévy[1], a este espaço se dá o nome de espaço cibernético (*cyberspace*), sendo um local para comunicação, sociabilidade, organização e transação, a formar assim, um nova fonte de informação e de conhecimento.

[1]Filósofo da informação que estuda as interações entre Internet e a sociedade (Tunísia, 1956)

2.2 Origem e Desenvolvimento da Internet

O Departamento de defesa dos Estados Unidos (*DoD – Departament of Defense*) ao final dos anos 60 criou a primeira rede nacional de computadores chamada de ARPANET (*Advanced Research Projects Agency Network*) que foi desenvolvida pela Agência de Pesquisas e Projetos Avançados (*ARPA – Advanced Research Projects Agency*) para garantia da segurança do país em caso de problemas nas comunicações.

Durante os conflitos da Guerra Fria contra a União Soviética, os Estados Unidos necessitavam desenvolver um sistema de comunicação que não fosse vulnerável a ataques. A ARPANET então foi criada financiada pelo governo e utilizava um conjunto de ligações centrais de uma rede com altas taxa de transferência de dados (*backbone*), que passava por baixo da terra para interligar os militares e pesquisadores. Dessa maneira, não possuía um centro definido para armazenamento de informações, a dificultar a sua localização e destruição.

A rede utilizava linhas telefônicas dedicadas com velocidade de 56Kbps (*kilobits por segundo*) e possuía elementos ativos denominados de dispositivos de processamento de mensagens (*IMP – Interface Message Processors*). Essa rede funcionava através de um protocolo de controle de rede (*NPC – Network Control Protocol*) e utilizava um sistema de transmissão de dados onde as informações eram divididas em pequenos pedaços. Esse processo denomina-se comutação por pacotes. Esses pacotes, por sua vez, continham trecho dos dados, endereço de destino e informações essenciais para remontagem da informação original [3].

Algumas instituições e universidades no início dos anos 70 foram autorizadas pelo DoD a se conectar a ARPANET. Essa liberação de acessos gerou uma grande expansão da rede para desenvolvimento de trabalhos cooperativos em diversas áreas de conhecimento, mas se constatou que seu sistema de protocolos era inadequado para ser utilizado em múltiplas interligações devido à grande quantidade de localidades inseridas na rede. Para suprir essa necessidade, foi criado um novo sistema de protocolos, o TCP/IP (*Transmission Control Protocol e Internet Protocol*), que se caracteriza como um conjunto de protocolos para conexão de sistemas heterogêneos [4].

A criação de um novo protocolo possibilitou o crescimento da rede. Com isso o governo americano divide a ARPANET em duas partes, uma com fins militares chamada de MILNET (*Military Network*) e outra para localidades não militares.

A fundação científica dos Estados Unidos (*NSF - National Science Foundation*), no final dos anos 70 desenvolveu uma sucessora para a ARPANET, a NSFNET (*National Science Foundation Network*). Esse projeto ocorreu devido ao grande crescimento das pesquisas universitárias nos Estados Unidos e a grande dificuldade de ingressar na ARPANET, que era controlada pelo DoD e apenas liberava acesso por meio de contratos de pesquisa [3].

A NSFNET teria como objetivo ser aberta a todos os centros de pesquisa universitários espalhados pelo país. A partir de então, construiu sua própria rede de *backbone* a interligar seis centros de supercomputadores. Cada supercomputador era ligado a um microcomputador, que por sua vez era conectado a linhas dedicadas de 56kbps, a formar uma sub-rede com tecnologia física proveniente da ARPANET e tecnologia lógica para comunicação com o propósito de utilizar o sistema de protocolos TCP/IP como padrão [3].

Com o financiamento de diversas redes, a NSF inseriu diversas universidades, museus, laboratórios de pesquisa e bibliotecas em sua rede com alta taxa de transferência de dados. A comunicação entre todos esses locais se dava através de ligação a um dos seus supercomputadores. A NSFNET fez sucesso rapidamente, mas logo se sobrecarregou devido à baixa capacidade de sua rede. Dessa forma, foi aberto espaço para diversas empresas investirem em capacidades maiores para a rede com uma única infra-estrutura que fosse competitiva e tivesse fins lucrativos.

Na década 80, o TCP/IP passou a ser a pilha protocolar padrão e as redes NSFNET e ARPANET sem fins militares foram interligadas e veio a criar o que hoje é conhecido como Internet. Essa interligação teve um grande crescimento e foram criadas conexões entre diversos lugares no mundo. Apenas na década de 90 a Internet deixou de ser voltada apenas para pesquisas e passou a ter um perfil comercial. Isso só foi possível devido a criação da aplicação Rede de Alcance Mundial (*WWW - World Wide Web*), desenvolvida por Tim Berners-Lee, um físico da Organização Européia para a Investigação Nuclear (*CERN - Conseil Européen pour la Recherche Nucléaire*). Ele também foi responsável pela criação do primeiro navegador de Internet (*Web Browser*) que consiste em um programa que permite interação entre utilizadores e documentos eletrónicos por meio do protocolo HTTP, descrito na Sessão 2.5.1.

A Web facilitou a inserção dos recursos da rede devido ao pioneirismo na aplicação de hipertexto para compartilhamento de recursos de informação. Ela foi introduzida em primeiro momento no CERN e, a princípio, disponibilizava informações através de textos. Somente após a associação da WWW com o Mosaic a Internet veio a entrar em popularidade. O Mosaic foi o primeiro navegador gráfico desenvolvido em 1993 por Marc Andressen no Centro Nacional de Aplicações de Supercomputação (*NCSA – National Center for Supercomputing Applications*). Junto com a Web tornou-se possível a configuração de páginas a conter textos, figuras, sons e até vídeos. Com isso diversos tipos de páginas foram criadas com os mais diversos tipos de conteúdo, como mapas, catálogos, indicadores financeiros, programas de rádios, etc. Com a chegada do século XXI a infraestrutura das redes se tornou presente em todos os lugares e com isso qualquer conteúdo ou serviço passou a estar disponível eletronicamente. Como exemplo, tem-se votação eletrónicas, comércio eletrónico, governo eletrónico e tantas outras aplicações passadas para vias digitais. Os computadores pessoais perderam espaço para dispositivos de mobilidade [13].

Fora a criação de navegadores gráficos, outro fator que foi fundamental para o expansão da Internet foi a criação de Provedores de Serviços de Internet (*ISP – Internet Service Providers*). Estas empresas oferecem aos utilizadores individuais acesso a Internet provendo diversos serviços agregados como *e-mail* e acesso à Web. Na década de 90, elas reuniram milhões de utilizadores a alterar todo o propósito da Internet, que passou de uma rede acadêmica de pesquisa e de proteção de informações militares, para um serviço de utilidade pública em todo o mundo. Atualmente pode ser acessada de qualquer lugar, a qualquer momento e quando o utilizador desejar em face aos serviços oferecidos [4].

O desenvolvimento nos últimos anos deixou a rede mundial de computadores acessível para praticamente todas as classes sociais. A influência de diversas culturas a tornou uma ferramenta de comunicação em massa.

2.3 Segurança da Internet

Durante o período inicial de utilização da ARPANET, a segurança da informação a ser transmitida não era prioridade. Isso porque o acesso era realizado em computadores de grande porte, mais conhecidos como *mainframes*, em ambientes restritos, controlados e com poucos utilizadores. A troca de mensagens por correio eletrónico e pesquisas computacionais foram as principais finalidades da rede.

Porém, entre as décadas de 60 e 70, utilizadores começaram a acessar informações de forma remota. Esse tipo acesso introduziu um novo risco aos dados mantidos remotamente, pois esses mesmos dados poderiam ser acessados por qualquer pessoa, autorizada ou não. O acesso físico aos terminais era supervisionado por um responsável de segurança que iniciava o processo de identificação e autenticação de cada utilizador. Como haviam poucos terminais, o processo de acompanhamento de utilização e atividades era simples. O fato de não existir políticas de segurança para aplicação de senhas de acesso mais elaboradas fez com que a quebra de sigilo e o compartilhamento de senhas entre os utilizadores se tornasse uma grande ameaça.

Com o conceito de multi-utilizador e sistemas de compartilhamento inseridos pela implantação de computadores pessoais, as redes de computadores de fato começam a existir. Assim, ocorre o surgimento dos controles de acesso para os utilizadores terem espaços de trabalhos independentes uns dos outros e sem nenhum tipo de interferência.

O modelo de segurança HRU (*Harrison, Ruzzo, Ullman mode*) [5] é um dos trabalhos pioneiros que atuavam a nível de sistema operacional e com a integridade dos direitos de acesso no sistema. Outros trabalhos que se destacam são: o modelo de confidencialidade *Bell-LaPaluda*, descrito como

uma máquina de estados finitos e foi desenvolvido para aplicar controle de acesso a informação [6], e o modelo *Diffie-Hellman* introduz o conceito de assinaturas digitais. Esse modelo por sua vez, estabelece um compartilhamento de chaves secreto que pode ser usado para troca de mensagens secretas dentro de um canal de comunicação público e inseguro [7].

Na década de 80, a utilização de computadores pessoais se intensifica e as primeiras aparições de vírus[2] são registradas. Em 1988 houve o primeiro registro de um *worm*[3], criado por Robert T. Morris Jr., que infectou em um curto período 10% dos computadores conectados a Internet. Dentre as redes afetadas, destacam-se as redes do Pentágono (sede do DoD dos Estados Unidos), da Agência Nacional de Segurança (*NSA - Nacional Security Agency*) e o Instituto de Tecnologia de Massachusetts (*MIT - Massachusetts Institute of Technology*). A principal funcionalidade desse *worm* era explorar as vulnerabilidades das conexões dos protocolos TCP e SNMP (*Simple Network Management Protocol)* de um sistema operacional Unix.

Esse evento se tornou significativo porque gerou um alerta na comunidade sobre aspectos de segurança e a necessidade de investimento em formas de conter e evitar esses problemas. O debate sobre segurança foi intensificado e vários países desenvolveram leis de punição aos atacantes, bem como ferramentas de proteção. Assim, surge o conceito restrição de acesso as redes chamado de *firewall*. Essa aplicação foi desenvolvida para isolar a redes locais (*LAN - Local Area Network*) e metropolitanas (*WAN - Wide Area Network*) através de um filtro com políticas de segurança, no conjunto de protocolos TCP/IP, na medida em os pacotes são transmitidos e recebidos [8]. No mesmo período também foram desenvolvidos programas para combater os vírus, chamados de antivírus.

O crescimento das LAN da WAN se intensificaram nos anos 90, período em que a Internet passa a ter milhares de utilizadores. Os ataques começaram a ser mais sofisticados e passaram a focar os pontos de acesso (*gateways*) entre os computadores e os servidores de conteúdo e serviços. No final da década são registrados os primeiros casos negação de serviço[4] (*DoS - Denial of Service*) e códigos maliciosos para roubo de informações dos utilizadores.

Com a chegada do século XXI a infraestrutura das redes se tornou presente em todos os lugares e com isso qualquer conteúdo ou serviço passou a estar disponível eletronicamente. Como exemplo, tem-se votações eletrónicas, comércio eletrónico, governo eletrónico e tantas outras aplicações passadas para vias digitais. Os computadores pessoais perderam espaço para dispositivos móveis

[2]Programas criados para realizar modificações prejudiciais aos computadores de forma a danificar sistemas, corromper ou destruir dados.

[3]Semelhante ao vírus, mas com a funcionalidade de se replicar. Através dessa característica ele cria cópias operacionais e infecta outros computadores pela rede local, Internet ou anexos de mensagens eletrónicas.

[4]Forma de ataque que torna indisponível os recursos de um sistema para os seus utilizadores

(*smartphones, notebooks, tablets, etc.*) e a computação móvel através de conexões sem fio (*Wi-Fi e Blueetooth*) se popularizou. Os sistemas de pagamento em tempo real (*on-line*) com a utilização de cartões de crédito também se intensificaram. Em contrapartida, todo esse desenvolvimento também trouxe inúmeras vulnerabilidades de rede de forma a comprometer a integridade das informações [9].

A utilização de analisadores de tráfego para realizar interceptação das comunicações foi desenvolvida em larga escala para uma compreensão em detalhes, do funcionamento da comunicação entre os dispositivos na rede. Agora era possível reunir informações vitais de desempenho dos dispositivos, do comportamento da rede, de possíveis falhas de serviços, priorização do tráfego por serviços mais sensíveis, quais serviços mais utilizados em determinado período e quais dados estão a ser transmitidos. Esses dados compõem um conjunto de informações de alto valor.

A convergência de diferentes conteúdos e serviços para o mundo digital fez com que a informação se transformasse em um ativo importante para o ambiente de negócios. Desta forma, é necessário que seja mantida em sigilo para preservar sua integridade e a privacidade de seus utilizadores. Em trabalhos como [10], fica claro a quantidade de informações que podem ser coletadas de uma rede e o quanto pode ser perigosa a exposição de todos esses dados. Sendo assim, a encriptação das informações passou a ser essencial para segurança das redes e seus utilizadores. Nos últimos anos a adoção de comunicação encriptada tem sido adotada em larga escala e cada vez mais a proteção dos dados tem sido uma temática tratada com muita atenção.

2.4 Arquitetura da Internet

A popularização da rede mundial de computadores tornou o TCP/IP a pilha protocolar mais usado em redes locais. Em relação a outros modelos de protocolos, o TCP/IP, tem a vantagem de ser roteável, ou seja, foi criado para redes de grandes e de longas distâncias, onde há vários caminhos para a informação atingir o seu destino [11]. Por possuir uma arquitetura aberta, o TCP/IP, pode ser modificado ou adaptado por qualquer fabricante. Por essa característica tornou-se um protocolo universal a dar possibilidade de interagir diversos sistemas sem problemas de comunicação.

Como mencionado anteriormente, o TCP/IP é formado por um conjunto de protocolos mais importantes, sendo um o TCP que tem função de transporte confiável fim-a-fim de mensagens de dados entre dois sistemas, o outro é o IP que é responsável pelo encaminhamento de pacotes de dados entre diversas sub-redes desde a origem até o destino [4]. O modelo TCP/IP foi decomposto em vários módulos que realizam tarefas específicas. As tarefas são realizadas em uma ordem precisa, sendo considerado um sistema estratificado, podendo ser dividido em camadas [12]. As

camadas desse modelo referencial são:

- camada de acesso à rede ou física;
- camada de rede;
- camada de transporte;
- camada de aplicação.

2.4.1 Camada de Acesso à Rede ou Física

É a primeira camada da pilha de protocolos e corresponde ao meio físico, ou seja, aos dispositivos (*hardwares*) envolvidos na comunicação entre as redes, onde os dados são tratados como pulsos elétricos. Tem como principal função a interligação do modelo TCP/IP com diversos tipos de redes e é responsável por enviar e receber dados em forma de pacotes, contendo endereço de origem, dados e o endereço de destino. Por existir uma grande variedade de tecnologias de rede a utilizar diversas velocidades, protocolos e meios de transmissão, a camada de acesso a rede não tem normatização que possibilita interconexões e inter-operações de redes heterogêneas [11]. Também é conhecida como a camada de enlace (*datalink layer*) devido a se encarregar do envio dos dados recebidos da camada de rede em forma de quadros através dos dispositivos da rede.

2.4.2 Camada de Rede

A camada de Rede é responsável pelo roteamento dos dados, endereçamento dos equipamentos na rede e é definida pelo protocolo IP [11]. Todo dispositivo de rede possui um endereço lógico atribuído a um número, denominado de endereço IP, que é associado a uma ou várias interface de um equipamento ou terminal (*host*) a identificar a rede e o próprio equipamento nessa rede. A principal função do protocolo de interconexão é fazer a transferência de blocos de dados, chamados de datagramas (*frame*). Baseia-se em um serviço sem conexão, ou seja, faz apenas o roteamento dos dados pela rede e não faz nenhum tipo de verificação de erros durante a transferência. Por causa desse serviço sem conexão, cada datagrama é considerado como uma unidade independente e tem comunicação não confiável, pois não utiliza nenhuma forma de reconhecimento fim-a-fim ou entre nós intermediários [3].

Também não existem mecanismos para controle de fluxo e de erros. Ocorre apenas uma simples conferência do cabeçalho para garantir que as informações contidas no mesmo sejam encaminhadas corretamente por *gateways*. Em uma arquitetura Internet TCP/IP é através do endereço IP que as estações conseguem enviar e receber mensagens pela rede.

2.4.3 Camada de Transporte

Esta camada está situada entre a camada de rede e a camada de aplicação e é responsável por receber os dados enviados pela camada de aplicação, que será vista no próximo tópico, e transformá-los em pacotes a serem repassados para a camada de rede através de um canal lógico de comunicação fim-a-fim. A arquitetura na camada de transporte baseia-se em um serviço de transporte orientado à conexão para garantir a entrega dos dados na ordem certa, fornecido pelo protocolo TCP. Mas também baseia-se em serviço de datagrama não orientado à conexão fornecido pelo protocolo UDP (*User Datagram Protocol).*

O protocolo TCP segmenta um fluxo de dados e os numera em determinada sequência que permita sua remontagem quando chegar ao destino. Antes de fazer uma transmissão o protocolo de uma estação de origem faz contato com o protocolo da estação de destino e estabelece uma conexão, que leva o nome de circuito virtual. Essa comunicação é chamada de orientada à conexão e precisa de uma confirmação do destinatário para ocorrer à troca de informações de maneira confiável [11]. O processo de confirmações é mantido durante todo o período em que a conexão ocorrer. Quando o envio dos segmentos termina, o protocolo TCP aguarda confirmação da estação de destino para finalizar a transmissão. Caso algum segmento não seja devidamente confirmado ele é retransmitido. Sendo assim, o protocolo TCP, é um protocolo orientado à conexão e altamente confiável [12].

O protocolo UDP, ao invés de fluxo, recebe blocos de dados de outras camadas. Da mesma maneira que o protocolo TCP, faz a segmentação e numeração dos blocos para permitir a reconstrução da informação no destinatário. Os segmentos não são sequenciados após serem numerados, pois o protocolo não dá importância à ordem de envio dos mesmos. Dessa maneira não existe uma confirmação de recebimento pela estação de origem e o protocolo UDP é dito como não-confiável. Também é considerado como não-orientado à conexão, pois não estabelece um circuito virtual para transferência dos blocos de dados. De maneira resumida pode-se dizer que o protocolo TCP é utilizado para transporte de dados de maneira confiável e o UDP para transporte rápido e não confiável de dados [12].

Para comunicação com camadas superiores, os protocolos TPC e UDP, utilizam portas que servirão para registro lógico de diferentes sessões estabelecidas simultaneamente através da rede. Existem 65.536 portas e estas são numeradas de 0 a 65.535. Cada porta corresponde a um serviço distinto ou pode ser usada por um programa, a tornar possível que uma estação utilize serviços em todas essas portas a aplicar um único endereço IP válido. São definidas por documentos que fazem a descrição de padrões de protocolos de Internet denominados de requisições de mudança (*RFC – Request for Comments*) e são de responsabilidade da autoridade para atribuição de números de Internet (*IANA - Internet Assigned Numbers Authority*), organização mundial encarregada de

coordenar alguns dos principais elementos que mantêm a Internet em funcionamento.

As portas reservadas de 0 a 1023 são para serviços mais conhecidos e utilizados como servidores de mensagens eletrónicas (porta 25), compartilhamento de arquivos (portas 20 e 21), etc. As portas maiores ou iguais que 1024 são usadas pelas camadas superiores para estabelecer conexões com dispositivos diversos a identificar a aplicação origem e destino de uma porta UDP ou TCP [12].

2.4.4 Camada de Aplicação

Camada responsável pela interação dos protocolos da camada de transporte e as aplicações que utilizam meios de comunicações de dados, ou seja, define os protocolos necessários para interligar as aplicações, fazer seu controle e as especificações dos dispositivos (interface) com o utilizador [11]. Esta camada não possui um padrão e cada aplicação possui seu próprio protocolo estabelecendo um padrão específico. Sendo assim, é formada pelos protocolos utilizados nas diversas aplicações do modelo TCP/IP. E ainda faz o endereçamento na rede através de portas de comunicações com a camada transporte, onde cada aplicação possui uma porta pré-definida, como já foi mencionado na tópico anterior. Alguns exemplos de protocolos dessa camada são:

- *FTP - File Transfer Protocol:* protocolo para serviços de transferências de arquivos;
- *SMTP - Simple Mail Transfer Protocol:* protocolo utilizado por servidores de mensagens para seu gerenciamento e distribuição;
- *HTTP - HyperText TransPort Protocol:* disponibiliza um dispositivo, normalmente um navegador, para visualização de páginas na Internet. Sendo implementado por um serviço Web;
- *DNS - Domain Name System:* protocolo que mapeia nomes para endereços IPs.

2.5 Protocolos de Comunicação

2.5.1 HTTP

É o protocolo de transferência mais empregado em toda Web e sua versão 1.1 é uma das mais utilizadas na Internet até os dias de hoje. É definido pelas RFCs 1945 e 2616 e utiliza formato de texto ASCII[5] (*American Standard Code for Information Interchange*) para comunicação. Esse

[5]Código apresentado por Robert W. Bemer como recurso de unificação para representação de caracteres alfanuméricos em computadores e se tornou a linguagem comum entre todos os dispositivos.

protocolo é executado por dois programas: um cliente, chamado de navegador Web (*User Agent*) e um servidor, chamado de servidor Web (*Web Server*). Os dois programas são executados em sistemas finais diferentes e conversam entre si por meio de troca de mensagens [3]. O HTTP determina a estrutura dessas mensagens e a forma como cliente e servidor interagem. Também é responsável pela forma que os clientes realizam a requisição das páginas aos servidores, que por sua vez, fazem a transferência aos clientes.

O HTTP trabalha na camada de aplicação para transferência de páginas HTML (*Hypertext Markup Languge*), que é uma linguagem de programação utilizada para desenvolver páginas Web e proporciona a elaboração de documentos que podem ser acessados e transmitidos por qualquer dispositivo pela Internet. Cada página pode ser constituída de muitos objetos, como imagens, animações e vídeos. Cada objeto, para ser acessado, tem seu próprio endereço eletrónico denominado de URL (*Uniform Resource Locator*). Assim, quando um utilizador solicita uma página Web, o navegador encaminha ao servidor Web as mensagens de requisições HTTP para os objetos de cada página. O servidor recebe as requisições, as processa e responde com mensagens de resposta HTTP com os objetos que foram solicitados. Essa interação pode ser verificada na Figura 2.1.

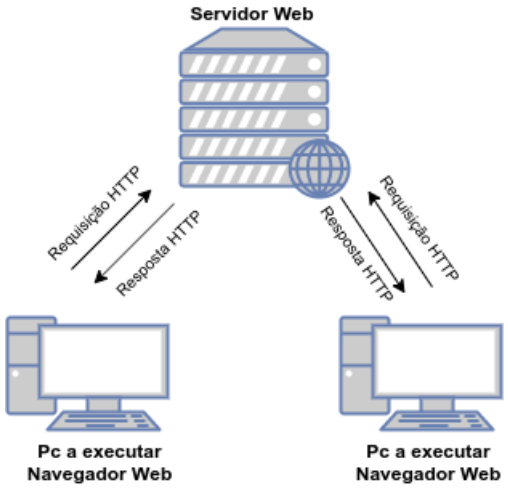

Figura 2.1: Interação cliente-servidor.

2.5.2 Conexões

O HTTP utiliza o protocolo TCP para transporte de dados pela Internet. Assim, um navegador para estabelecer contato com o servidor, envia uma conexão TCP para o servidor através de uma porta de comunicação 80. O TCP é um protocolo orientado à conexão e ao fluxo, pois realiza a interconexão de sistemas e constrói uma conexão estável entre computadores remotos com a

confirmação de entrega das mensagens através da Internet. Com essas características, nem os navegadores e nem os servidores ficam responsáveis com mensagens perdidas, duplicadas, longas ou qualquer tipo de confirmação.

Também considera-se o HTTP como um protocolo sem estado (*stateless*), ou seja, cada solicitação realizada pelo cliente é independente. Isto significa que assim que o navegador encerra à conexão TCP, não existe registo de informação. Além de todas as requisições serem independentes, não possuem conhecimento uma das outras.

Quando um cliente realiza um pedido de uma página web e recebe-o de volta existe um tempo para que essa solicitação ocorra. Esse tempo é chamado de RTT (*Round Trip Time*), que é o período que um pacote leva para ir do cliente ao servidor e de volta ao cliente. Em algumas situações o RTT pode incluir atrasos na transmissão dos pacotes por meio de dispositivos intermediários e em seu tempo de processamento.

Diante desses casos, o HTTP pode adotar dois tipos de conexões:

- *conexões não persistentes:* ao proceder uma transferência de um objeto entre o cliente e o servidor, à conexão TCP é fechada e não fica disponível para outros objetos da página Web. A cada solicitação é necessário um nova conexão TCP, que exige comunicação de três etapas (*three-way handshake*) devido à necessidade de reconhecimento entre cliente e servidor para comunicação real dos dados. Na primeira etapa o cliente envia uma mensagem com segmento TCP ao servidor. Em um segundo momento, o servidor reconhece o segmento e envia uma resposta ao cliente. Por último, o cliente confirma a resposta enviada pelo servidor e como à conexão TCP encontra-se pronta para o transporte de dados, também envia uma requisição HTTP para receber a página Web e os objetos solicitados. A depender da aplicação utilizada, esse tipo de conexão pode comprometer seu funcionamento com eventuais atrasos;

- *conexões persistentes:* à conexão TCP fica ativa e disponível para envio de qualquer solicitação de objetos pelo cliente. A conexão se encerra após um período de tempo sem requisições.

2.5.3 Formatos de Mensagem HTTP

As mensagens pode ser de dois tipos: requisição (*request*) e resposta (*response*).

Requisição

Essa mensagem é dividida em três partes: linha de requisição, linha de cabeçalho e corpo da mensagem. A primeira parte é a linha de requisição que é o pedido que cliente faz ao servidor e possui os campos URL para localizar o endereço dos objetos solicitados, a versão do HTTP e o método. O método se refere a algum tipo ação que a mensagem demanda, como solicitar ou preencher informações, e é bastante utilizado em mensagens HTTP. Os métodos que podem ser operados são:

- *GET:* requisita a leitura de uma recurso Web;
- *HEAD:* regressa os cabeçalho de uma resposta;
- *POST:* envio de recursos a uma página Web.
- *PUT:* requisita armazenamento de uma página Web;
- *DELETE:* remoção de recursos.
- *TRACE:* verifica se todas as solicitações estão sendo processadas pelo servidor;
- *OPTIONS:* forma de consulta do cliente sobre propriedades do servidor ou de objetos específicos;
- *CONNECT:* facilita a comunicação entre servidores intermediários com recursos de comunicação segura;
- *PATCH:* aplica alterações parciais de recursos.

A segunda parte da mensagem corresponde às linhas de cabeçalho que possuem os detalhes das solicitações ao servidor. O cabeçalhos podem ser de três tipos:

- *gerais:* contêm as informações sobre a própria mensagem e o servidor as utiliza para para controlar seu processamento e notificar o cliente com informações extras;
- *de requisição:* além de dar controle ao cliente de como as requisições são processadas e fornecer informações mais detalhadas ao servidor, pode informar quais formatos ou códigos que o cliente pode verificar;
- *de entidade:* se existir, descrevem de forma breve o conteúdo da mensagem.

O corpo da mensagem é a terceira parte que constitui uma mensagem de requisição e trará uma entidade, que pode ser um arquivo de imagem, uma página Web ou qualquer outro objeto.

Resposta

As mensagens de resposta também são divididas em três partes: linhas de estado, linhas de cabeçalho e o corpo da mensagem. As linhas de estado informam a versão HTTP utilizada na resposta, um código de resposta de três dígitos entre servidor e cliente e uma mensagem associada ao código que em um texto descritivo para identificação das respostas do servidor. A tabela 2.1 descreve alguns exemplos de códigos e suas respectivas mensagens.

Tabela 2.1: Códigos e mensagens HTTP.

Código	**Mensagem**
200 OK	Requisição efetuada com sucesso
301 Moved Permanently	Objeto solicitado foi removido
400 Bad Request	Servidor não processou a solicitação do cliente de forma correta
404 HTTP Not Found	Servidor não encontrou o objeto solicitado
505 HTTP Version Not Supported	Servidor não suporta a versão HTTP solicitada

As linhas de cabeçalho que possuem os detalhes das solicitações ao servidor. Os cabeçalhos podem ser de três tipos:

- *gerais:* contêm as informações sobre a própria mensagem, de forma análoga nas mensagens de requisição;
- *resposta:* informações complementares as linhas de estado e notificação de erros;
- *de entidade:* são mais frequentes em mensagens de reposta e também descrevem de forma breve o conteúdo da mensagem.

O corpo da mensagem, da mesma forma que nas mensagens de requisição, trará uma entidade, que pode ser também um arquivo de imagem, uma página Web ou qualquer outro objeto.

A Figura 2.2 demonstra um exemplo das mensagens de requisição e resposta.

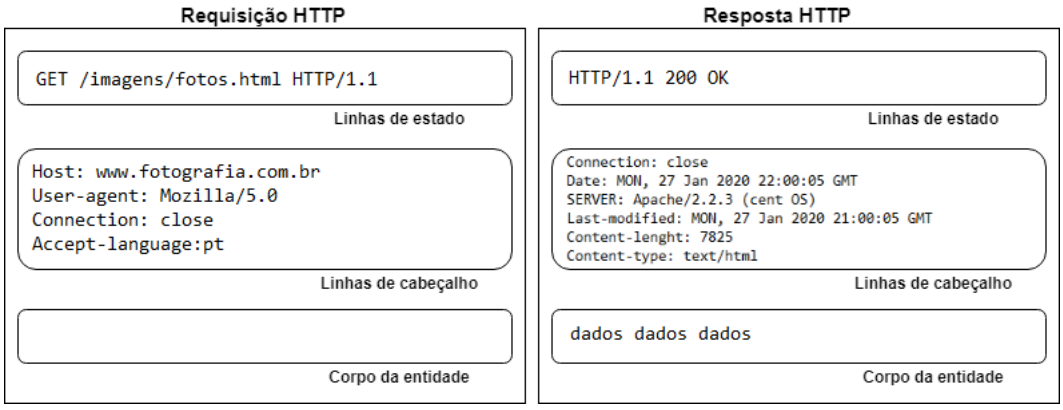

Figura 2.2: Mensagens de requisição e resposta do HTTP.

2.5.4 Cookies

Como informado anteriormente, o HTTP não possui estado e as sessões de solicitação e resposta não são armazenadas. Nos primórdios da Web, essa ferramenta era adequada às necessidades da época. Porém, com a inserção de serviços seguros, a necessidade de autenticação dos utilizadores passou a ser essencial devido a questões de segurança. Atualmente, os servidores precisam de algum recurso que exiba o conteúdo correto a cada utilizador.

Uma técnica, conhecida como *cookies* e formalizada na RFC 2109, foi proposta para enviar arquivos de texto simples do servidor Web ao Navegador. Esse arquivo é enviado no primeiro acesso a uma nova página Web e armazena os dados do utilizador, para que em um próximo acesso, suas informações sejam configuradas automaticamente. Com as informações salvas, toda vez que o utilizador acessar uma página Web, o servidor saberá direciona-lo corretamente ao seu conteúdo ou serviços. O tempo de armazenamento dessas informações é determinado pela página Web e pode ter validade de dias ou anos.

A utilização de *cookies* possibilitou a facilidade da navegação, mas também provoca alguns problemas. A princípio, eles deveriam apenas ter propriedades de comunicação com a página Web para qual foram originados. No entanto, são alvos de ataques de interceptação de dados devido ao alto valor das informações de cada utilizador. Além de informações privativas, como números de cartões de crédito, é possível reunir informações de navegação, o que possibilita um mapeamento de preferências de cada utilizador.

Essas informações em larga escala podem ser utilizadas de diversas formas, pois ela se consolidou como um ativo que potencializa os lucros em diversos negócios. Grandes corporações como Google, Amazon e Facebook exploram essa utilização. Existe uma discussão sobre a distribuição,

de forma lícita ou não, desses dados. Diante desse cenário, muitos países aderiram a Lei Geral de Proteção aos Dados[6] com intuito de proteção do tratamento da informações de cada utilizador.

2.5.5 Cache Web

Cache Web ou *proxy* é um servidor que responde a requisições HTTP em nome de um servidor Web de origem [4]. Isto é, o servidor Web principal tem as solicitações gravadas e disponibilizadas em servidores intermediários com a finalidade de possibilitar um acesso mais eficiente em relação aos recursos de rede (tráfego, largura de banda, latência, etc.) e processamento de requisições no servidor (recursos físicos). Muitos ISPs utilizam esse serviço para aceleração de acesso dos seus utilizadores e de espelhamento dos servidores Web para melhoria de desempenho. A Figura 2.3 ilustra um servidor cache entre o utilizador e o servidor Web.

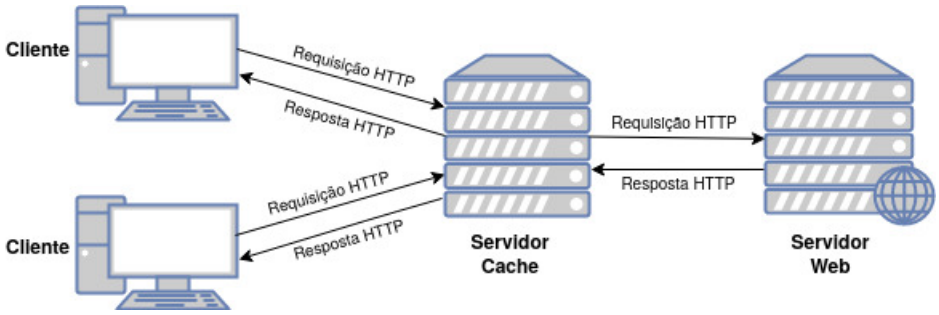

Figura 2.3: Servidor cache Web.

2.5.6 Versões HTTP

O HTTP se tornou o protocolo mais utilizado na Web e sua versão 1.1 está ativa há mais de vinte anos. Durante a sua criação, a preocupação de implantar opções para uso futuro fez com o protocolo acompanhasse durante um bom período a evolução da Web. Ao longo do tempo, várias das implementações previamente propostas foram pouco utilizadas. Quando passaram a operar geraram diversos problemas de interoperabilidade [13].

Um exemplo claro destas funcionalidades é o *pipelining*. Técnica na qual o HTTP envia várias requisições simultaneamente, por conexão TCP, sem aguardar as respostas correspondentes. Para que essa função opere corretamente as respostas devem retornar na ordem que foram solicitadas.

[6]Regulamentação de armazenamento e gestão de dados dos utilizadores por empresas e organizações. Disponível em *https://eur-lex.europa.eu/legal-content/PT/TXT/PDF/*.

Se um servidor Web tiver problemas com a latência da rede e atrasar o processamento de todas as solicitações enviadas em paralelo, poderá ocasionar respostas fora de ordem e gerar alguns problemas [13]. Por exemplo, em um acesso a uma determinada página Web, o cliente sofrerá uma lentidão no carregamento de todos os objetos. Essa função está desativada por padrão nos navegadores modernos.

A versão 1.1 do HTTP ainda se depara com outras limitações:

- linhas de cabeçalho longas, que ocasionam atraso no carregamento de páginas Web;
- sensibilidade a latência, o que leva a não utilizar todo potencial do TCP;
- não realiza multiplexação de várias solicitações em uma única sessão TCP.

HTTP/2

Em 2015, a RFC 7540, propôs uma nova atualização ao HTTP que recebeu o nome de HTTP/2. Essa versão foi derivada do protocolo experimental SPDY (*speedy*) criado pela Google. O SPDY é um protocolo não padronizado que foi desenvolvido para transporte de dados com redução de latência no carregamento de páginas e segurança no acesso a Internet. De uma forma geral, possui o funcionamento muito similar ao HTTP.

O HTTP/2 busca corrigir os problemas encontrados no HTTP/1.1, ser compatível com as aplicações atuais e trazer novas funcionalidades:

- compatibilidade total com o TCP para requisição/resposta entre cliente e servidor, de forma a manter o padrão do HTTP;
- multiplexação de várias sessões TCP em uma única conexão;
- priorização de fluxo de dados;
- um algoritmo que permita que o cliente e servidor decidam por utilizar a versão 1.1 ou 2.0 do HTTP, já que uma transição de protocolos em servidores poderá levar alguns ano;
- compatibilidade com métodos, códigos e campos do cabeçalho do HTTP/1.1;
- permite compressão de cabeçalhos;
- possibilita envio de conteúdo ainda não solicitado pelo cliente;
- suporte a utilização do protocolo em navegadores de computadores pessoais, dispositivos móveis, servidores Web, servidores *proxy*, *firewalls* e redes de entrega de conteúdo;

- mais eficiência para diminuir latência e melhorar velocidade de carregamento de páginas Web;
- possui encriptação por padrão, para maior segurança entre páginas.

Muitos navegadores modernos já possuem suporte e compatibilidade a essa versão do HTTP, mas mesmo com novas propostas, não é um protocolo mais veloz e nem o mais moderno. Esse fato se dá principalmente por manter a compatibilidade com a versão 1.1 e os mesmos métodos para requisição. A compressão dos cabeçalhos é vulnerável a vários tipos de ataque como *BREACH*, que busca interceptação das respostas HTTP, e *CRIME* que é o sequestro de uma sessão através das informações de *cookies* [14].

Outra questão importante são as configurações de criptografia, que também foram mantidas as mesmas da versão 1.1 e possibilita que servidores Web possam escolher níveis mais baixos de segurança a deixar utilizadores em risco. Como forma de correção, os desenvolvedores dos navegadores realizam ativação do protocolo HTTP/2 somente se houver disponibilidade dos protocolos SSL/TLS.

HTTP/3

Ainda não foi oficializado como um novo padrão de protocolo para a Internet. Porém, o IETF[7](*Internet Engineering Task Force*) tem atualizado um documento preliminar (*Internet Draft*[8]) com validade até julho de 2020, a descrever especificações preliminares, tendo como resultados e pesquisas sobre essa nova versão.

O HTTP/3 deixará de utilizar o TCP como padrão e passará a adotar o QUIC (*Quick UDP Internet Connections*), protocolo proposto pela Google e concebido para oferecer confiabilidade, segurança e redução de latência no transporte de dados [15]. O QUIC estabelece novas funcionalidades na camada de transporte, em nível superior, sobre o protocolo UDP e propõe-se a disponibilizar todos os propósitos e garantias fornecidos pelo TCP, de forma superar suas limitações e melhorar sua eficiência.

A seguir se apresentam algumas das características propostas para a nova versão do HTTP:

- os fluxos serão compartilhados, em uma mesma conexão através do protocolo QUIC entre cliente e servidor, e enviados independentemente aos utilizadores. Essa funcionalidade não afetará o recebimento dos dados se houver perda de um único pacote;

[7]Comunidade internacional composta de desenvolvedores, operadores e pesquisadores que se dedicam a evolução da arquitetura e funcionamento da Internet.

[8]Disponível em https://tools.ietf.org/html/draft-ietf-quic-http-25.

- requisições a endereços sem comunicação de segurança (*http://*) não serão mais suportadas. Passará a adotar endereços seguros (*https://*) e criptografia através do protocolo TLS;
- fará controle de fluxo no nível da conexão com limitação de *buffer*[9] que uma transmissão pode consumir e não comprometer o envio dos dados de outros fluxos;
- realizará detecção e controle de perdas dos dados através de estimativas de RTT;
- redução drástica de RTT no carregamento de páginas Web.

2.6 Protocolos de Comunicação Segura

2.6.1 SSL

O SSL (*Secure Socket Layer*) permite tráfego seguro pela Internet e é um protocolo proposto pela Netscape Communications, uma empresa americana desenvolvedora de serviços independentes para dispositivos em 1994. Atualmente está na versão 3.0, proposta na RFC 6101, e tem como principal atributo o fornecimento de privacidade e confiabilidade entre dois aplicativos de comunicação. Através de uma camada de proteção adicional de criptografia na conexão do protocolo TCP, entre utilizador e o servidor Web ao qual ele está conectado. O protocolo SSL assegura uma transmissão de dados sigilosa e anônima.

Os dados da camada de aplicação antes de serem enviados para a camada de transporte são passados para a camada SSL, onde ocorre o processo de criptografia dos dados. Após o encapsulamento dos dados, um cabeçalho SSL é adicionado o qual irá permitir que cliente e servidor realizem um processo de autenticação e negociação da conexão. A autenticação identificará a legitimidade do utilizador no servidor, enquanto a negociação se encarregará da escolha de um algoritmo de criptografia e chaves criptográficas antes que o protocolo da aplicação transmita ou receba seu primeiro *byte* de informação [16]. Após as informações estarem codificadas, são passadas para a camada de transporte que faz a adição de seu próprio cabeçalho e passa para a camada Internet. Por último, os dados são enviados pela camada física até o destino.

No processo de recepção no dispositivo de destino, os dados são recebidos pela camada física, depois pela camada de rede, pela camada de transporte onde são decodificados e os dados originais são transmitidos para a aplicação correspondente. A Figura 2.4 demonstra o funcionamento de uma sessão SSL.

[9] O conteúdo transmitido após ser dividido em pacotes, é armazenado numa área temporária da memória dos dispositivos, para posteriormente ser acessada pelo utilizador ou pela aplicação.

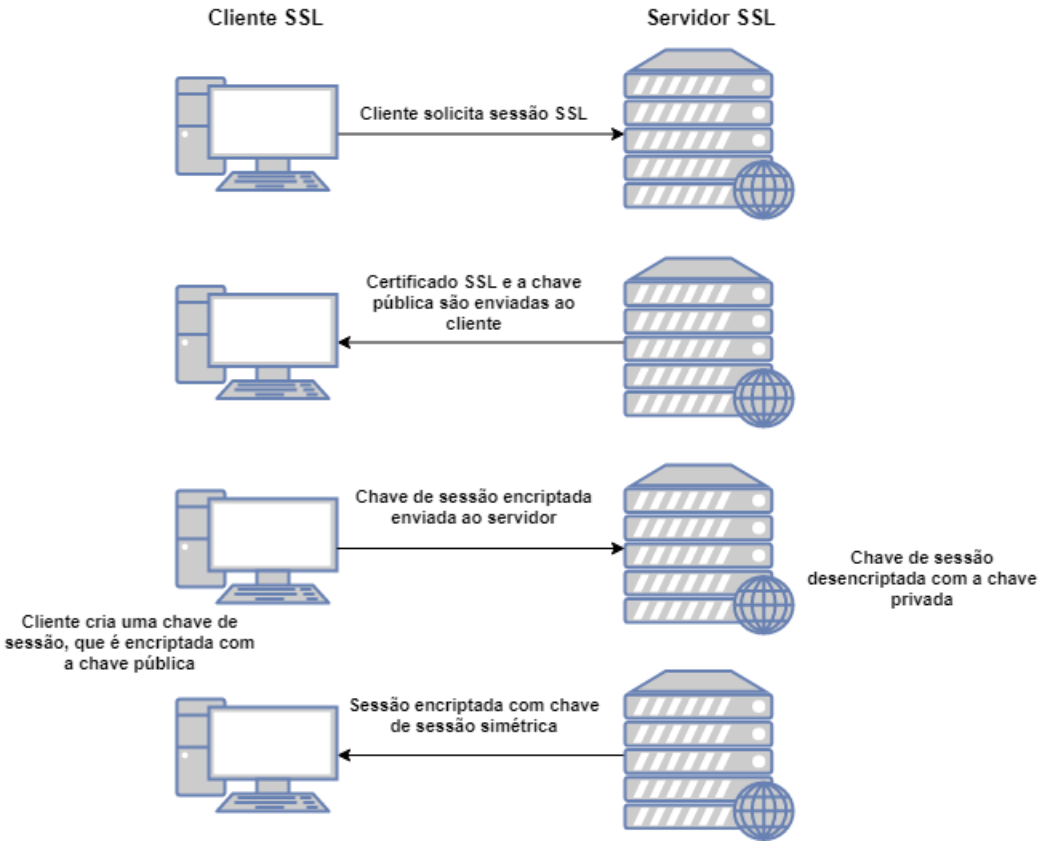

Figura 2.4: Sessão SSL.

2.6.2 TLS

Outro protocolo de segurança para promover privacidade e integridade das informações trocadas entre dois aplicativos de comunicação. Atualmente o TLS (*Transport Layer Security*) está na versão 1.3 proposta na RFC 8446, mas foi introduzido inicialmente em 1999 pelo IETF. O TLS é uma versão que propõe proteção a qualquer comunicação entre navegadores e servidores Web, como serviços de correio eletrónico, mensagens em tempo real e voz sobre IP (*VoIP*).

O TLS é composto por duas camadas: *Record Protocol* e *Handshake Protocol*. A primeira camada é responsável por fornecer conexão privada através de criptografia simétrica, que utiliza algoritmos com a mesma chave de criptografia para encriptar o texto pleno e decriptar o texto cifrado. As chaves geradas nesse processo são exclusivas de cada conexão e se baseiam em um segredo negociado por outros protocolos, como o próprio TLS *Handshake Protocol* [17]. Outra responsabilidade da primeira camada é confiabilidade de conexão através de verificação de integridade das mensagens por códigos de autenticação (*MAC - Message Authentication Code*).

A segunda camada, *Handshake Protocol*, viabiliza a segurança da conexões através da autenticação por criptografia assimétrica, que opera com uma chave de criptografia secreta e outra pública. Esse processo pode ser opcional, mas em geral é necessário estar ativo em um dos lados da conexão. Um outra característica dessa camada é de não disponibilizar de nenhuma forma o segredo negociado entre dois dispositivos a fazer com que ele não seja interceptado. Dessa forma, como nenhum atacante pode pode modificar a comunicação de negociação sem ser detectado, à conexão com TLS é dita como confiável [17].

Por estar acima da camada de transporte, o TLS, trabalha de forma independente dos protocolos da camada de aplicação. Porém, é necessário que os desenvolvedores implementem segurança aos protocolos escolhidos para trabalhar em conjunto ao TLS. Porém, o processo de *handshake* do TLS é complexo e composto de uma grande quantidade de mensagens entre cliente e servidor, o que pode impactar o tráfego e o desempenho da rede.

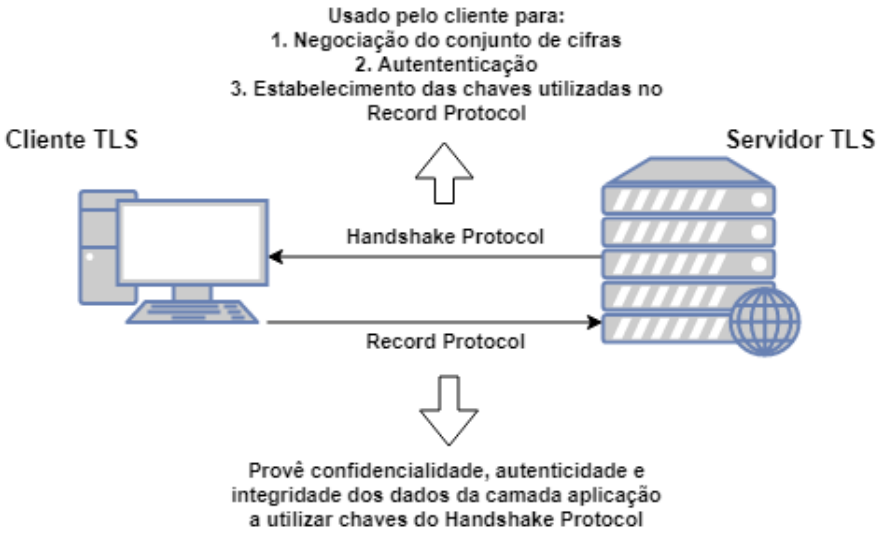

Figura 2.5: Sessão TLS.

2.6.3 HTTPS

O HTTPS (*Hypertext Transfer Protocol Secure*) é a versão do protocolo HTTP com uma camada de segurança adicional que emprega os protocolos SSL/TLS e foi proposto na RFC 2818. O HTTPS, através da camada de segurança, permite que dados sejam enviados por uma conexão criptografada onde a autenticidade entre o cliente e servidor seja verificada por certificados digitais [4].

Os certificados digitais, por sua vez, são documentos eletrónicos que são aplicados para identificar um utilizador na Internet. Portanto, são considerados como identidade virtual e têm a finalidade de garantir a autenticidade, confidencialidade, integridade e segurança dos dados em qualquer operação realizada em ambiente virtual [16]. Cada utilizador é associado a um certificado de chave pública emitido por uma autoridade de certificação (*CA - Certification Authority*), que são instituições responsáveis pela emissão de certificados confiáveis.

A finalidade do HTTPS é de impossibilitar que a informação transmitida seja interceptada por terceiros. Um utilizador pode identificar na barra de endereços, em seu navegador, a presença de um símbolo em forma de cadeado que constata que a página Web foi certificada como segura. Na Figura 2.5 é possível ver um exemplo no navegador Chrome de uma conexão segura HTTPS, onde a presença de um certificado representado por um cadeado fechado, constitui a relação de confiança entre o navegador e o servidor Web.

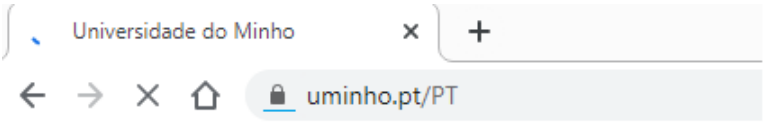

Figura 2.6: Exemplo de navegador a utilizar HTTPS.

Uma conexão HTTPS ocorre de forma simples:

1- servidor envia chave pública juntamente com um certificado de validação da mesma para o utilizador;

2- utilizador consulta a CA para verificar estado do certificado, se está válido ou revogado;

3- se o certificado estiver válido, o navegador confia na chave pública e estabelece comunicação com o servidor Web de forma legítima;

4- se o certificado estiver revogado, o navegador não conseguirá estabelecer comunicação com o servidor Web.

Conforme mencionado anteriormente, o HTTPS só difere do HTTP pela camada de segurança e não possui modificações nas funcionalidades gerais do protocolo. A Figura 2.7 demonstra a diferença entre a utilização dos protocolos HTTP e HTTPS.

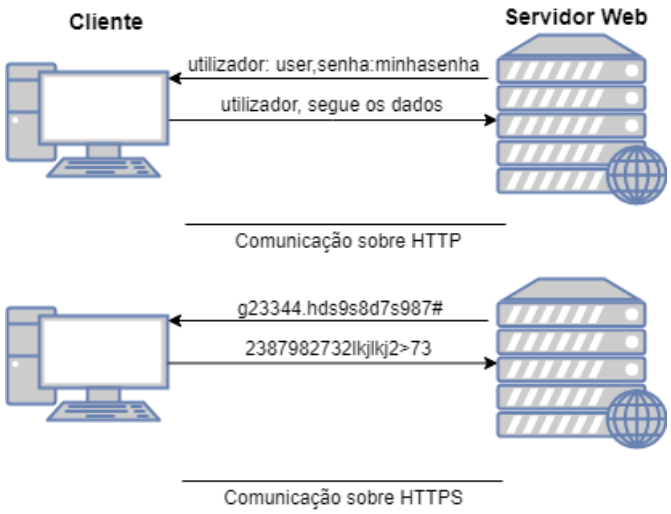

Figura 2.7: Diferença entre HTTP e HTTPS.

2.7 Classificação do Tráfego de Rede

A classificação do tráfego modificou a compreensão das redes de computadores, pois possibilitou mensurar diversas caraterísticas na procura de falhas e melhorias de desempenho. A evolução da Internet trouxe imensos desafios devido a muitas aplicações buscarem não serem detectadas, seja para proteção dos dados de seus utilizadores, seja para utilização de recursos de rede sem nenhum tipo de controle.

Por outro lado, pesquisadores, operadores de redes e grandes ISPs necessitam dessas informações para conhecer as particularidades do tráfego, gerenciamento de recursos, segurança da informação e cobrança de utilização de serviços baseados no consumo. Essas duas visões de utilização da análise de tráfego proporcionou uma variedade de técnicas de classificação.

2.7.1 Técnicas de captura do tráfego

Técnicas Router-Based

Técnica baseada nas informações retiradas através das funções de um roteador (*router*). Muitas vezes pode ser uma aplicação como o *Netflow*, que é um recurso dos roteadores da fabricante Cisco e tem como finalidade coletar informações nas interfaces de entrada e saída de dados. Assim, administradores de rede são capazes de executar análises de origem e destino do tráfego, classe de serviços e possíveis causas de congestão.

Outra forma de retirar informações é através do protocolo SMNP (*Simple Network Management Protocol*) que é um protocolo de gestão de recursos para Internet. Adotado na camada de aplicação, foi concebido para ser um protocolo padrão implementado por todos os fornecedores e esta disponível na maioria dos dispositivos que se conectam a uma rede [4]. O SNMP fornece recolhimento de informações de todos os dispositivos conectados, onde pode se extrair informações vitais para a administração da rede. Também possui um ferramenta complementar a sua base de dados, chamada de RMON (*Remote Network Monitoring*), que se caracteriza como uma norma padrão para facilitar o monitoramento da rede através de dispositivos remotos. Além de apresentar métodos para configuração e controle da rede, possibilita a coleta de dados a partir de dispositivos de monitoramento.

Técnicas Non Router-Based

São técnicas baseadas em coletar informações diretamente dos sistemas através de dispositivos ou aplicações específicas. Pode ser de dois tipos:

- *medições ativas:* realizada mediante a injeção de tráfego na rede. Isto é, insere tráfego de prova para para verificar o comportamento da rede em termos de conectividade, desempenho e disponibilidade de serviços oferecidos. É uma técnica que permite aos administradores de rede e pesquisadores realizarem testes mais específicos as funcionalidades da rede. Em relação a outras técnicas, captura uma quantidade menor de dados e pode ocasionar impactos negativos na rede, já que pode causar sobrecarga de recursos no tráfego existente;
- *medições passivas:* realizada para registro do tráfego sem causar interferência na rede. Essa técnica utiliza instrumentos para capturar informações no momento em que circulam pela rede e mostra-se bastante efetiva devido a possibilidade de coletar um volume de dados bastante representativo.

As técnicas de medição podem ser caracterizadas em dois tipos: centralizadas, quando as informações após coletadas são enviadas para um servidor para análise e armazenamento, ou distribuídas que é quando as informações são coletadas em pontos críticos da rede.

2.7.2 Métodos de Classificação

Segue uma breve definição dos métodos mais utilizados na classificação do tráfego:

- *classificação baseada em portas:* um dos métodos pioneiros na classificação que analisa portas de origem/destino e os protocolos utilizados durante o processo de comunicação.

Mas no processo evolutivo da Internet, esse método tornou-se ineficaz devido à diversas aplicações adotarem técnicas de disfarce ou portas aleatórias;

- *classificação baseada na carga útil (payload):* denominado de inspeção profunda de pacotes (*DPI - Deep Packet Inspection*) é método para investigação do conteúdo dos pacotes. Através de informações existentes no campo de *payload* da camada de transporte faz comparação com uma base de dados para dedução do tipo de tráfego [18]. No entanto, se o tráfego estiver encriptado resulta numa solução ineficiente;

- *classificação baseada em recursos do fluxo:* surgiu como alternativa a DPI e emprega o recurso de análise de dados automatizados para construir modelos analíticos recorrendo, por exemplo, a técnicas de *machine learning*. Esse método compromete-se em classificar aplicações através de dados estatísticos para identificar padrões no fluxo de informações [19];

- *classificação comportamental de sistemas terminais:* método baseado no comportamento dos sistemas que fazem a recepção do fluxo de dados. Essa classificação é composta de três níveis e proposta por [20]: o primeiro nível verifica a popularidade de acesso ao terminal face ao número de conexões existentes. O segundo nível se encarrega de descobrir se o terminal é um servidor ou consumidor de serviços ou colabora com outras conexões. Por último, o terceiro nível se responsabiliza em capturar as comunicações da camada de trasporte para identificar o aplicativo de origem.

2.8 Sumário

Esse capítulo apresentou uma visão geral do processo evolutivo da Internet e de como a segurança tornou-se fundamental após seu desenvolvimento. Também abordou o funcionamento da Internet através da pilha protocolar TCP/IP, bem como o principal protocolo utilizado para comunicação na camada de aplicação, o HTTP. Além da evolução do protocolo, abordou-se sua aplicação com camada de segurança e os principais protocolos utilizados para essa fim. Por último, foi realizado um resumo sobre as principais técnicas de análise e medição do tráfego e seus métodos de classificação. Algumas das ferramentas existentes para identificar e classificar o trafego, e que foram utilizadas neste trabalho, serão comentadas na sessão 4.1.2.

Capítulo 3

Critérios de Pesquisa e Trabalhos Relacionados

Nesse capítulo é descrito brevemente o processo de recolha de material de referência por intermédio de uma revisão sistemática da literatura (*SLR - Systematic Literature Review*). Esse tipo de metodologia faz uso de critérios específicos e possibilita a identificação e avaliação das melhores fontes de informações relativas ao tema a ser desenvolvido. Também são apresentados alguns artigos, resultados da SLR, que se relacionam diretamente com a temática abordada e servem para elucidar diversos assuntos relativos aos impactos do HTTPS na navegação do utilizador final.

3.1 Revisão Sistemática da Literatura

Desde que a Internet foi concebida a necessidade de modificação, para acompanhamento do desenvolvimento tecnológico, foi essencial. Porém, essas mudanças geraram impactos negativos e positivos entre a interação cliente/servidor, ou seja, na navegação Web em geral. A SLR, por sua vez, teve papel fundamental em mapear artigos onde esses impactos estivessem relacionados ao utilizador final.

Baseado em [22] e para para cobrir a bibliografia base de forma efetiva, a metodologia SLR foi composta de cinco etapas: definição das questões de pesquisa (*research questions*) que servem de referência a este trabalho, determinar as fontes de pesquisa e criação de um termo de busca (*research query*) com palavras chaves, seleção de artigos relevantes, estudo dos artigos selecionados e extração/síntese dos dados encontrados. A figura 3.1 demostra a metodologia adotada.

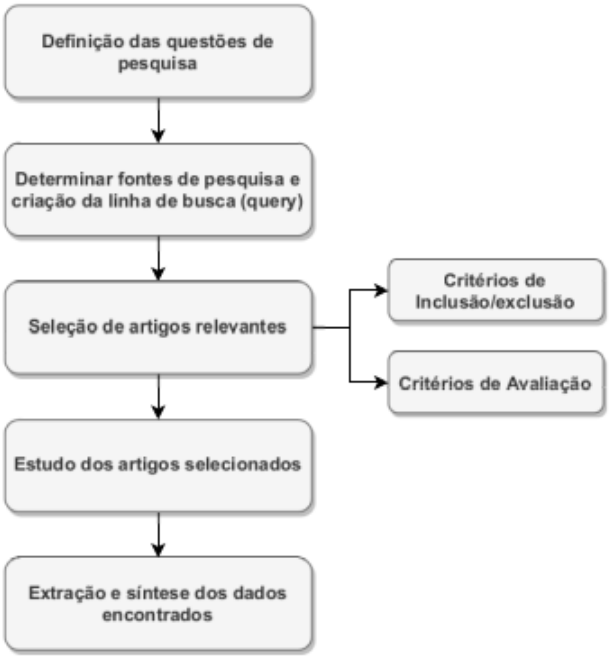

Figura 3.1: Metodologia SLR.

3.2 Questões de Pesquisa

Algumas *research questions* foram desenvolvidas para restringir um campo de análise específico, que se refere diretamente ao impacto da navegação do utilizador final:

1- Qual é o impacto da utilização da criptografia na navegação Web do utilizador final (segurança e latência na navegação)?

Com a evolução da Internet a necessidade de proteção dos dados se tornou fundamental. Além da preservação da identidade do usuário, a segurança da informação é essencial para diversas aplicações via Web, que vão desde consultas até transações bancárias. Essa questão tenta buscar os impactos na navegação do utilizador final, principalmente em relação à latência, já que medidas preventivas de segurança foram inseridas na estrutura da rede.

2- Como se efetua e se estrutura a comunicação HTTPS com base nos protocolos SSL/TLS?

Com a necessidade de mais segurança na navegação, a criação de protocolo para gerar comunicação segura foi essencial. Diante disso, essa questão irá abordar como o HTTPS se estrutura e sua relação com os protocolos SSL/TLS.

3- Que tipo de informação é possível extrair do HTTPS que seja relativa à comunicação cliente-servidor?

Para verificar modificações na forma de navegação do utilizador é imprescindível identificar quais os campos essenciais de informação presentes no protocolo de comunicação.

4- Qual a percentagem de Navegação Segura é obtida pelos utilizadores com recurso aos vários dispositivos, como computadores pessoais e dispositivos móveis?

Segundo algumas plataformas de informação, como o relatório de transparência do Google [1], retratam em pontos percentuais a quantidade de endereços que realmente trabalham com encriptação dos dados. Essa questão tratará de identificar se de fato a segurança está implantada no nível informado por essas ferramentas.

5- Qual é a política de Navegação Segura usada pelos grandes fornecedores de conteúdos e serviços (Google, Yahoo, Bing, Facebook, Amazon,etc.)?

Cada provedor de serviços possui uma política própria voltada para segurança. Assim, essa questão é necessária para esclarecer se as diferenças geram algum tipo de impacto ao utilizador final.

6- Que ferramentas de análise de tráfego cifrado existem? Que técnicas e algoritmos são usados para análise de tráfego cifrado?

A classificação do tráfego precisou se moldar aos avanças de tecnologia no campo de encriptação dos dados. Atualmente existem várias formas de análise e essa questão trará uma visão geral das ferramentas existentes e quais serão utilizadas neste trabalho.

7- Que evolução e modificações se efetuaram na Navegação Segura nos últimos anos?

Essa questão visa reunir as modificações do processo de navegação segura proveniente da adoção da navegação encriptada ao longo dos últimos anos.

3.3 Termos de Busca

Com as questões de pesquisa definidas, uma *research query*, foi desenvolvida segundo os parâmetros definidos em [22]. Esse método consiste em derivar as questões de pesquisa em palavras chaves, identificar sinónimos para cada termo principal, construir uma sequência de pesquisa utilizando as formas boolianas, com o **OR** para relacionar sinónimos encontrados e o **AND** para relacionar os termos de pesquisa principais.

Todos os termos foram utilizados em inglês, devido ser a língua considerada como universal e

o idioma com a maior quantidade de trabalhos relevantes publicados. Em relação ao critério de escolha das palavras chave, foram selecionados cinco temas principais e seus sinónimos com maior referência ao tema de comunicação segura entre cliente/servidor. Os termos principais definidos estão listados a seguir e a Tabela 3.1 representa o resultado final do método aplicado.

- *network:* representa suporte para as palavras chave referentes a comunicação entre dois ou mais dispositivos em rede;
- *security:* representa suporte para as palavras chave associadas a segurança na Web;
- *traffic classification:* representa suporte para as palavras chave relativas à caracterização do tráfego;
- *search engines:* representa suporte para para as palavras chave direcionadas as principais motores de busca em relação a navegação segura de seus utilizadores;
- *devices:* representa suporte para as palavras chave inerentes a navegação segura em sistemas operacionais para qualquer tipo de dispositivo.

Tabela 3.1: Termos de busca e principais sinónimos.

Termos	**sinónimos**
Network	client-server communications, HTTP, latency, Web Content Provider
Security	HTTPS,TLS, SSL, secure web browsing, cryptography, security police
Traffic classification	traffic analysis, payload extraction, DPI, LPI
Search engines	Google, Yahoo, Bing, DuckGo, Gibiru
Devices	Android, Linux, Windows

Com todos as palavras chaves escolhidas, uma *research query* foi desenvolvida de forma genérica para poder ser aplicado em qualquer base de dados e poder apresentar resultados consistentes:

("secure web browsing" **OR** HTTPS **OR** TLS **OR** SSL **OR** cryptography) **AND** ("traffic analysis" **OR** "client-server communications" **OR** latency **OR** "payload extraction" **OR** "deep packet inspection" **OR** DPI **OR** "light packet inspection" **OR** LPI) **AND** ("web content provider" **OR** "security police") **AND** ("mobile device" **OR** desktop **OR** Android **OR** Linux **OR** Windows) **AND** (Google **OR** Facebook **OR** Yahoo **OR** Bing **OR** DuckGo **OR** Gibiru).

3.4 Fontes de Pesquisa

As conferências TMA *(Network Traffic Measurement and Analysis Conference)* e PAM *(Passive and Active Measurement Conference)*, serviram de marco inicial para uma pesquisa dos termos principais de busca das palavras chaves, pois são conferências exclusivas e de vasto conteúdo da área de pesquisa de modelação e caracterização do tráfego.

Através dessas conferências foi possível localizar outras conferências relevantes à mesma área, bases de dados *on-line* relevantes, portais de buscas de trabalhos acadêmicos e portais relacionados diretamente a área de análise de tráfego. A Tabela 3.2 lista as principais fontes que foram utilizadas como base de pesquisa.

Tabela 3.2: Fontes de pesquisas.

Locais de Pesquisa	**Fontes**
Base de dados online	IEEE Xplore, Web Of science, ACM Digital Library
Motores de busca online	Google Scholar, Microsoft Academic
Conferências	TMA, PAM, INFOCOM, SIGCOMM, SIGMETRICS, ACM IMC
Outras fontes	CAIDA

Das fontes de pesquisas selecionadas é importante ressaltar que:

- a base de dados IEEExplorer disponibiliza acesso a artigos, jornais, conferências publicadas pelo Instituto de Engenheiros Elétricos e Eletrónicos (*IEEE - Institute of Electrical and Electronics Engineers*). Dentre as conferências escolhidas, o IEEE já inclui em sua base de consulta a INFOCOMM (*International Conference on Computer Communications*);
- a biblioteca digital ACM *(Association for Computing Machinery Digital Library)* é uma base de dados *on-line* que concentra todos os artigos publicados pela própria ACM, revistas, jornais, livros e conferências. Dentre as conferências escolhidas para serem fontes de pesquisa, a ACM já inclui em sua base de consultas: SIGCOMM (*Special Interest Group on data Communications*), SIGMETRICS (*Special Interest Group on Measurement, Evaluation and Modeling of Computer Systems*) e IMC (*Internet Measurement Conference*);
- conferências escolhidas por alta relevância na área de análise de tráfego são: TMA e PAM;
- a CAIDA (*Center for Applied Internet Data Analysis*) foi adicionada as buscas porque é um centro de pesquisa com o objetivo de efetuar pesquisas em redes, desenvolver infraestruturas para suporte a coleta da informações em larga escala, reunir e distribuir dados para a comunidade científica [23].

3.5 Critérios de Avaliação

De acordo com a *research query* citada no item anterior, 170 trabalhos foram encontrados com temática relacionada ao tema principal deste trabalho. Além de ser uma quantidade muito grande de trabalhos, nem todos possuem relação direta com o impacto da navegação encriptada para o utilizador final. Sendo assim, foram escolhidos alguns critérios de inclusão e exclusão para refinar a pesquisa realizada.

- *critérios de inclusão:* se o artigo estiver publicado em conferências e jornais entre os anos de 2015 e 2020. Se o artigo estiver correlacionado diretamente as áreas: de classificação de tráfego, de protocolos de comunicação e de protocolos de comunicações com segurança. E se o artigo tiver acima de 5 citações;
- *critérios de exclusão:* artigos que não considerarem HTTP, HTTPS, SSL e TLS. Artigos que não abordam encriptação e análise de tráfego. Artigos correlacionados a temática de navegação segura (*Web safe*[1]) ao invés de segurança na Web (*Web security*).

Após aplicar os critérios de inclusão e exclusão houve uma redução no número de artigos que passou para 40, conforme mostrado na Tabela 3.3. Porém, a quantidade continuava alta e tornou-se necessário trabalhar em novos critérios que assegurassem maior qualidade dos trabalhos pesquisados. Baseado em [24], foram criados novos parâmetros avaliativos através de portais de classificação referentes a qualidade das conferências e jornais onde os artigos foram publicados, e também da conteúdo dos artigos em relação ao tema do trabalho.

Sendo assim, três portais de classificação foram utilizados para atribuir pesos aos níveis de avaliação de cada publicação:

- CORE (*The Computing Research and Education Association of Australasia*) é uma associação dos departamentos universitários de ciência da computação na Austrália e na Nova Zelândia que realiza a classificação das principais conferências da comunidade acadêmica. A base de dados de classificação do CORE encontra-se disponível em *http://portal.core.edu.au/conf-ranks/*;
- QUALIS é um sistema brasileiro que classifica a produção científica e é mantido pelo sistema CAPES (*Coordenadoria de Aperfeiçoamento de Pessoal de Nível Superior*) associado ao Ministério da Educação do Brasil. Possui sua base de dados de classificação disponível em *https://sucupira.capes.gov.br/sucupira/public/index.xhtml*;

[1]Trabalhos relacionados a forma segura de acesso a Internet pelo utilizador focada em atualizações de segurança de antivírus, sistemas operacionais, etc.

- SCImago (*SCImago Journal Country Rank*) é um portal disponível ao público que inclui os jornais e indicadores científicos de países desenvolvidos a partir das informações contidas no banco de dados Scopus[2]. Esses indicadores podem ser usados para avaliar e analisar domínios científicos de diversas áreas. Sua base de dados encontra-se disponível em *https://www.scimagojr.com.*

Tabela 3.3: Artigos encontrados e selecionados em cada fonte de pesquisa.

Fonte	**Artigos Encontrados**	**Artigos Selecionados**
IEEExplorer	50	21
Web Of Science	35	2
ACM DL	37	4
Google Scholar	21	5
Microsoft Academic	19	0
Conferências	6	4
CAIDA	2	1
Total	170	40

Os critérios escolhidos de acordo com as classificações das conferências e jornais foram:

- *condição 1 (C1):* faz parte do grupo das conferências com classificação A ou A* no CORE, dos Jornais A de qualquer classificação no Qualis e dos jornais Q1 no SCImago. Essa classe possui peso 2;

- *condição 2 (C2):* faz parte do grupo das conferências com classificação A e A* no CORE, dos Jornais B de qualquer classificação no Qualis e dos jornais Q2 no ScImago. Essa classe possui peso 1.5;

- *condição 3 (C3):* faz parte do grupo das conferências com classificação B no CORE, dos Jornais B de qualquer classificação no Qualis e nos jornais Q3 no SCImago. Essa classe possui peso 1.0;

- *condição 4 (C4):* faz parte do grupo das conferências com classificação C no CORE, dos Jornais C de qualquer classificação no Qualis e nos jornais Q4 no SCImago. Essa classe possui peso 1.0;

[2]Considerado como um dos maiores banco de dados de resumos e citações da literatura com revisão por pares: revistas científicas, livros, processos de congressos e publicações do setor. Encontra-se disponível em *https://www.elsevier.com/pt-br.*

- *condição 5 (C5):* sem classificação atribuída no CORE, no Qualis e no SCImago. Essa classe possui peso 0.

Os critérios escolhidos de acordo com as qualidade dos artigos em relação ao tema abordado foram:

- *condição 1 (P1):* se faz parte da alguma conferência e/ou de algum jornal;
- *condição 2 (P2):* se o artigo aborda algumas das *research questions*;
- *condição 3 (P3):* se o artigo aborda o campo de análise e classificação de tráfego;
- *condição 4 (P4):* se o artigo cita os protocolos HTTP, HTTPS, SSL e TLS voltados para segurança na Web;
- *condição 5 (P5):* se o artigo contém informações claras e objetivas;
- *condição 6 (P6):* se o artigo possui citações.

  Todos os critérios voltados para classificação dos artigos possuem três estados:

  - atende as condições de classificação totalmente, este estado possui peso 1.0;
  - atende as condições de classificação parcialmente, este estado possui peso 0.5;
  - não atende as condições de classificação, este estado possui peso 0.

Com todos os critérios de avaliação aplicados aos artigos encontrados, dos 40 trabalhos, 18 satisfazem totalmente as condições de classificação propostas no item anterior. Na Tabela 3.4 é possível identificar quais *research questions* que são respondidas por cada artigo, assim como cada peso atribuído referente as classificações de cada conferência, jornais e conteúdo. Essa classificação tem como principal objetivo garantir a alta qualidade dos artigos encontrados e que sejam totalmente relevantes a área de análise e classificação do tráfego diante da crescente adoção de tráfego encriptado e seus impactos no utilizador final.

3.6 Resumo dos Trabalhos Selecionados

Liu et al. [25]: Os autores buscam identificação do tráfego HTTPS para os navegadores mais utilizados do mercado e para as páginas Web encriptadas mais utilizadas de acordo com a

Tabela 3.4: Artigos selecionados.

		Research Questions							Peso das Fontes	Peso
Estudo	**Referência**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**de Pesquisa**	**dos Artigos**
E1	Liu et al.	x	x	x	x		x	x	0.5	1
E4	Gill et al.	x	x	x	x	x		x	1.0	1
E5	Sapathy et al.	x	x		x	x			0.5	1
E6	Sung-Min et al.	x	x	x		x	x		0.5	1
E8	Durumeric et al.	x	x	x	x	x		x	1.5	1
E9	Muehlstein et al.	x	x	x	x	x	x	x	1.5	1
E12	Fang et al.	x	x	x	x	x	x	x	1.5	1
E14	Manzoor et al.	x	x	x	x		x	x	1.0	1
E15	Wijnants et al.	x	x	x	x	x	x	x	2.0	1
E17	Husák et al.	x	x	x	x	x	x		1.5	1
E19	Felt et al.	x	x	x	x	x	x	x	2.0	1
E21	Anrbak et al.	x	x	x	x	x		x	2.0	1
E23	Nayloret al.	x	x	x	x				2.0	1
E25	Gonzalez et al.	x	x	x	x				2.0	1
E26	Velan et al.	x	x	x			x	x	1.5	1
E37	Kausar et al.	x			x	x	x		1.5	0.5
E39	Khater et al.	x		x	x	x	x	x	1.5	1
E40	Finsterbusch et al.	x			x	x	x	x	1.5	1

classificação da Alexa[3]. Para essa identificação ser possível, é proposta a identificação HTTPS baseada no tamanho do pacote. Isto é, uma análise da possibilidade de usar o tamanho do pacote de solicitação para identificar o navegador com a criação de dicionários que realizem a distinção de fluxo de tráfego desconhecido. Desta forma, através de um algoritmo baseado em árvores de decisão é possível identificar o tamanho do pacote que cada navegador possui.

Gill et al.[26] : Estudo que relata a evolução da Web em termos tecnológicos em uma janela de 10 anos (1998-2008). Através da análise de *datasets*, mostra que era possível realizar uma investigação minuciosa e identificar serviços utilizados por um utilizador, servidores que proveem esses serviços, acessos que colocam a rede em risco, quais são os consumidores e a evolução do protocolo HTTP. Esse trabalho não leva em consideração a encriptação dos dados, mas mostra que é possível que através de requisições HTTP pode-se realizar diversos tipos de mapeamentos.

Sapathy et al. [14]: O estudo identifica que falhas de segurança colocam em risco os canais de comunicação e que é necessário assegurar tráfego seguro das aplicações fim-a-fim. Mediante a este cenário, é abordada uma revisão da estrutura dos protocolos SSL/TLS e suas principais

[3]Endereço eletrónico que reúne métricas de monitoramento de popularidade acesso de endereços em detrimento a outros endereços. Pode ser acessado por https://www.alexa.com

vulnerabilidades, seguido de uma pesquisa dos principais tipos de ataques conhecidos e o que influencia suas ocorrências.

Sung-Min et al. [27]: Apresenta um novo método para produção de assinaturas, de forma automática, de serviços a partir do *payload* dos protocolos SSL/TLS. A criação das assinaturas se destina a realizar a classificação do tráfego de rede através de seus serviços de aplicação. É explorado o campo de informações de publicação do certificado que se encontra no registro de troca de certificados do tráfego SSL/TLS para assinaturas de serviço. Assim, como o processo de *handshaking* ocorre antes da criptografia da transmissão é possível coletar as informações de sessão e endereço IP dos utilizadores e servidores para gerar identificadores baseado em assinaturas e baseados em endereços encontrados. O estudo informa que os resultados experimentais possuem 90% de precisão pra todos os serviços SSL/TLS.

Durumeric et al. [28]: Estudo que informa os resultados de uma medição do ecossistema do protocolo HTTPS em larga escala. Esse ecossistema abrange a infraestrutura da chave pública, que é adotada na maioria das comunicações seguras na Web. Também é realizada uma investigação nas relações de confiança entre as autoridades geradoras de certificados, autoridades intermediárias e os certificados usados pelos servidores Web. O estudo também relata que mais de 1.800 entidades podem emitir certificados atestando a identidade de qualquer endereço eletrónico enfraquecendo a segurança dos certificados.

Muehlstein et al. [29]: Trabalho que, através de técnicas de *machine learning*, considera cenários de ataques passivos a rede. Realiza a identificação do sistema operativo, navegador e aplicativos do utilizador a partir do tráfego HTTPS. Com a precisão de 96,06%, este trabalho fornece exemplos e novos recursos para explorar o comportamento das sessões SSL/TLS e dos navegadores mais utilizados atualmente.

Fang et al. [30]: Propõe um algoritmo de tempo real para reconstruir interações usuário-navegador. Esse algoritmo possui várias características modernas de tráfego Web para dispositivos móveis para identificar as solicitações de cliques, com precisão, de um conjunto de *streaming* intenso via HTTP. O estudo foi realizado através de um *dataset* de uma operadora móvel chinesa. É um estudo que serve de base para oferecer suporte, a qualquer operadora de telefonia móvel, e compreensão as características do tráfego da Web móvel e o comportamento dos assinantes.

Manzoor et al. [31]: O artigo, através de medição passiva e da utilização de ferramentas de DPI, realiza caracterização do tráfego HTTP/1 e HTTP/2. Esse estudo mostra a adoção crescente do protocolo HTTP/2 e compara os serviços atuais nas duas versões. Também cria uma metodologia para identificar o tráfego HTTP/2 em fluxos em que nenhuma informação da camada de aplicação está disponível.

Wijnants et al. [32]: Detalha uma profunda pesquisa sobre a implementação dos navegadores

modernos e os fatores que estão a levar a priorização do HTTP/2 como principal protocolo de comunicação. O artigo demonstra que a priorização desse tipo de tráfego não é trivial, devido a sua complexidade, e que alguns navegadores levam mais de 25% a mais do tempo médio para carregar uma página Web.

Husák et al. [33]: Demonstra um experimento, baseado no monitoramento de rede e impressão digital do protocolo SSL/TLS, para realizar um processo identificação leve e de tempo real de clientes HTTPS. O trabalho demonstra a possibilidade de estimar o navegador de um cliente que utiliza comunicação HTTPS por meio do *handshake* SSL/TLS. Essa estimativa torna-se possível porque cada navegador tem valores diferentes para as informações de cabeçalho HTTP. A partir dessas informações, dicionários foram criados para identificar navegadores e conexões em tempo real com precisão de 95,4%.

Felt et al. [34]: Estudo que reúne métricas para avaliar a adoção e impactos do protocolo HTTPS na perspectiva do cliente/servidor na Web. Para realizar essa avaliação, as métricas foram coletadas dos utilizadores agregados em larga escala dos navegadores *Google Chrome* e *Mozilla Firefox*. O trabalho também avalia o crescimento da adoção do HTTPS e identificação das áreas em que podem ter impactos relevante no ecossistema HTTPS.

Anrbak et al. [35]: Retrata as vulnerabilidades estruturais do HTTPS, mapeia o mercado de certificados e analisa as soluções regulamentares e tecnológicas sugeridas pelas organizações responsáveis. As pesquisas demonstram que o mercado HTTPS não difere do setor financeiro devido a apresentar incompatibilidades de informações e diversas carências, já que os certificados digitais são produzidos por um número restrito de entidades certificadoras. O artigo também estima que que as propostas de melhorias estão longe de ser adotadas em larga escala e que as falhas persistirão nos próximos anos.

Nayloret al. [36]: Por intermédio de *datasets* de grandes *ISPs* o estudo examinou resultados de uma adesão acelerada do HTTPS em um período de três anos. É mostrado que o HTTPS pode inserir mais custos na infraestrutura da rede, aumento de latência e mais consumo de dados e energia. Porém, a falta de clareza da comunicação criptografada resulta na ineficiência de visibilidade dos serviços que tenham valor agregado na rede de forma a dificultar seus custos reais . Esse artigo busca estimular a discussão sobre tecnologias que possam mitigar os custos do HTTPS e, ao mesmo tempo, proteger a privacidade do utilizador.

Gonzalez et al. [37]: Com a premissa de que o HTTPS deveria tornar o mapeamento de perfis de utilização mais difícil para qualquer pessoa além dos *endpoints* de comunicação, este trabalho examina até que ponto essa afirmação tem veracidade. Em primeiro momento é mostrado que mediante a indicação do nome do servidor do protocolo TLS ou DNS, que um atacante pode ter informações básicas para criação de um perfil de utilizadores de domínios com conteúdos homo-

gêneos. Já para conteúdos com grande variedade de direcionamentos, como portais de notícias e compras, essa técnica possui falhas. Ainda assim, a possibilidade de criar perfis de acesso precisos por pela impressão digital de tráfego que utiliza assinaturas de rede é totalmente viável para identificar a página exata em que um utilizador estar a navegar. Isso torna-se viável devido a impressão digital da camada de transporte permanecer forte e escalável.

Velan et al. [10]: Esta pesquisa tem como principal objetivo o fornecimento de uma visão geral e comparativa dos métodos para classificação e análise do tráfego encriptado. O estudo se divide em quatro etapas. Na primeira etapa descreve vários dos protocolos de criptografia mais utilizados com a demonstração de sua estrutura de pacotes e de seu comportamento padrão em uma rede para formar uma base de dados. Já na segunda etapa realiza uma investigação das informações fornecidas por cada protocolo de criptografia, onde os dados monitorados revelam cifras fracas. A descrição da estrutura dos protocolos de criptografia é utilizada na terceira etapa para detectar protocolos em uma rede através de algoritmos de classificação de tráfego. Na quarta e última etapa é fornecida uma diversidade de métodos baseados em comportamento para a classificação do tráfego encriptado, onde é possível obter informações muito detalhadas com destaque, em alguns casos, o conteúdo da conexão.

Kausar et al. [38]: Trabalho que apresenta as problemáticas da análise de tráfego e evidencia ataques para explorar vulnerabilidades encontradas nos dispositivos móveis, principalmente em *smartphones*. Esses dispositivos se tornaram essenciais para navegação diária de um utilizador e não estão livres de riscos à segurança, mesmo com o tráfego encriptado. Desta forma, é proposto um ataque de tempo (*timming attack*) que rastreia o tempo de conexão entre cliente e servidor para cada solicitação realizada pelo utilizador. Nesse tipo de ataque os recursos de tempo são observados e cada conexão tem um tempo diferente, o que permite que a conexão seja identificada com precisão de 96%.

Khater et al. [39]: Este trabalho aborda uma revisão crítica do campo de análise de tráfego de rede em tempo real através da utilização de algoritmos de *machine learning* para classificar o tráfego encriptado da Internet. Também é um trabalho que demonstra como os pesquisadores estão a buscar alternativas aos desafios atuais da classificação do tráfego encriptado.

Finsterbusch et al. [40]: Este artigo apresenta pesquisa que realiza uma análise completa e minuciosa dos mais importantes módulos de DPI de código aberto. Essa análise compreende uma avaliação da precisão da classificação, por meio de um conjunto comum de rastreamentos de tráfego com base realística e de seus requisitos computacionais. Também é apresentada uma avaliação técnica dos módulos de DPI e o estudo dos resultados obtidos que proporcionam uma proposta de diretrizes gerais para o desenvolvimento e implementação de módulos de DPI mais adequados.

3.7 Sumário

Este capítulo expôs de forma sucinta o processo de uma revisão sistemática de literatura para seleção de trabalhos relevantes na área de análise e classificação do tráfego. É descrito o processo de criação das *research questions* e *research queries* para direcionamento específico do campo de pesquisa, bem como a utilização de ferramentas para classificar a qualidade de cada trabalho. Também é abordado a adoção de critérios de qualidade utilizados para um distinção de conferências e jornais com maior avaliações no campo de pesquisas científicas. Por fim é realizado um breve resumo dos artigos selecionados por serem mais relevantes para a temática de protocolos de comunicações seguros e tráfego encriptado.

Diante dos trabalhos escolhidos, é possível notar que muitas das pequisas são direcionadas a buscar formas de identificar as mudanças na rede mundial de computadores após as implementações de criptografia. Diversos artigos tem o propósito de mensurar os efeitos do HTTP e suas novas versões, dos impactos do HTTPS, informações da camada de segurança e identificação do *payload* encriptado, seja voltada para o cliente, seja voltada para o servidor.

Porém, são poucos trabalhos que visam discutir o que modificou para o utilizador no âmbito do que é perceptível por ele. Isto é, com todas essas alterações, o utilizador final é realmente capaz de observar o que alterou em seu acesso? Será que o seu acesso se tornou mesmo seguro? Quais os efeitos o HTTPS trouxe para suas informações na Web?

Através desses questionamentos esse trabalho realiza um estudo para verificar, pela ótica do utilizador final, quais impactos o HTTPS trouxe para sua experiência de navegação a Internet.

Capítulo 4

Metodologia

A primeira parte do capítulo descreve o processo de criação de um ambiente de prova para verificação do protocolo HTTPS. Nele pretende-se simular, de forma fiável à realidade, a navegação de um utilizador comum em sua residência. Também são descritas todas as ferramentas utilizadas, a forma como os dados foram capturados e o processo de reprodutibilidade dos testes. A segunda parte aborda a metodologia utilizada no processamento dos dados colhidos. Desta forma, busca-se demonstrar quais funcionalidades específicas de cada ferramenta foram selecionadas para investigar em detalhe os impactos da implementação da segurança na Web na perspectiva de um utilizador comum.

4.1 Ambiente Experimental

Para análise dos impactos da utilização do protocolo HTTPS pelo utilizador final, é necessário definir um ambiente experimental de testes fiável a realidade. Isto é, um ambiente que ofereça condições de simular a navegação real de um utilizador em sua residência de modo a identificar, pelo tráfego gerado, o que a utilização de um protocolo seguro pode apresentar. A primeira parte desta sessão é responsável por descrever as principais etapas de conceção desse ambiente e foi dividida em duas sub-sessões:

- *infraestrutura:* corresponde à infraestrutura utilizada para criação e operação das máquinas virtuais, seus respectivos sistemas operativos, navegadores e conteúdo a ser acessado;
- *ferramentas de captura e testes:* descreve as ferramentas que necessitam estar instaladas nas máquinas virtuais para efetuarem o processo de captura do tráfego produzido. Também aborda os requisitos necessários para o processo de otimização e realização dos testes.

A segunda parte com as demais sub-sessões, são responsáveis por descrever o plano dos testes que aborda como foi realizada a sequência dos testes, a programação de suas funcionalidades de rotina de repetição e otimização, bem como o processo de medição dos dados. Também discute detalhes importantes dos navegadores e motores de busca utilizados, pois são de extrema importância para a compreensão da interação com protocolos de comunicação segura e seu possível comportamento durante a medição dos dados. Por último ocorre a demostração dos elementos necessário para reprodução do ambiente de testes.

4.1.1 Infraestrutura

Conforme mencionado, no processo de criação do ambiente de simulação foi necessário definir qual conteúdo seria acessado, quais navegadores seriam utilizados e quais sistemas operativos comportariam a experimentação.

Com o auxilio da ferramenta *Alexa Traffic Rank*[1], que monitoriza os acessos a uma página Web e realiza uma comparação da sua popularidade em relação a outras páginas, criou-se uma relação das URLs para acesso à Internet. Esses endereços foram separados por géneros: notícias, redes sociais, serviços multimédia (*streaming*) de áudio e vídeo, serviços de *e-mail*, e motores de busca. Os critérios de escolha dos endereços foram baseados em:

- maior número de acessos para endereços de notícias:
  - **www.whiplash.net:** é um dos maiores portais sobre música da América Latina e está em atividade há mais de 24 anos;
  - **www.uol.com.br:** é o maior portal de notícias do Brasil com mais de 113 milhões de visitantes únicos por mês e 6,7 bilhões de páginas visitadas mensalmente;
  - **www.nexojornal.com.br:** veículo de jornalismo eletrónico brasileiro e está entre os melhores portais de jornalismo independente do Brasil. Também é reconhecido por não exibir publicidade em seu conteúdo.
- maior diversidade de conteúdo produzido pelos utilizadores para redes sociais:
  - **www.instagram.com:** umas das principais redes sociais de compartilhamento de fotos e vídeos entre seus utilizadores. Além de permitir filtros digitais, possibilita a interação com outros endereços de serviços como Facebook, Twitter e Tumblr.
- principais plataformas de produção de conteúdos de áudio e vídeo para serviços multimédia:

[1] A ferramenta está disponibilizada no endereço https://www.alexa.com/topsites.

  - **www.soundcloud.com:** uma plataforma para publicação de áudio utilizada por profissionais de música. É um dos endereços mais utilizados para disponibilizar *podcasts*[2];
  - **www.youtube.com:** uma das maiores plataformas de compartilhamento de vídeo. É o segundo endereço eletrónico mais acessado pelos utilizadores na Internet.

- serviço de e-mail gratuito e com garantia de criptografia ponta a ponta:
  - **www.protonmail.com:** serviço de e-mail gratuito que utiliza criptografia para proteção de conteúdos e informações de seus utilizadores. Diferente de outros provedores de e-mail, esse servidor pode ser acessado através de um navegador, da rede Tor[3], além de ter compatibilidade com aplicativos de diversos sistemas operativos.

- motores de busca mais acessados e que garantam redirecionamento para endereços seguros e proteção de informações de utilizadores:
  - **www.google.com:** motor de busca mais utilizados na Internet e que garante o redirecionamento das pesquisas para endereços seguros;
  - **www.bing.com:** o segundo motor de busca mais utilizado na Internet. Além de ter sido desenvolvido pela Microsoft, também garante o redirecionamento de pesquisa para endereços seguros;
  - **www.search.yahoo.com:** o terceiro motor de busca mais utilizado na Internet e que também garante redirecionamento de pesquisas para endereços seguros;
  - **www.duckduckgo.com:** motor de busca alternativo que tem como principal filosofia a privacidade e o não registro de informações do utilizador;
  - **www.gibiru.com:** outro motor de busca alternativo que também não registra informações e não divulga informações do utilizador para endereços terceiros.

A opção de escolha dos sistemas operativos foi baseada na popularidade entre os utilizadores e na total compatibilidade com bibliotecas de automatização de testes. Como primeiro sistema operativo optou-se pelo Xubuntu, que é uma distribuição Linux e umas das versões de *software* livre mais utilizadas. O segundo sistema escolhido foi o Windows, devido a ser o principal sistema operativo do mercado de utilizadores finais nos últimos anos. Os sistemas foram emulados a partir de máquinas virtuais criadas através do programa *Virtual Box*[4], que é uma ferramenta específica

[2]Publicação de aquivos de áudio em formato digital, semelhante a uma transmissão de rádio. O conteúdo é criado sob demanda e fica disponível para o utilizador escutar quando quiser.

[3]Serviço de anonimização que tem como objetivo proteger as informações dos utilizadores quando elas chegam na Internet através de servidores de hospedagem ocultos.

[4]Mais detalhes da ferramenta podem ser encontrados em https://www.virtualbox.org.

para a virtualização de sistemas. Essa aplicação permite simular ambientes diferentes que um utilizador pode possuir em seu computador pessoal.

O processo de seleção dos navegadores também foi realizado através da popularidade de acesso entre utilizadores, através do último relatório comparativo da revista digital *Computer World*[5], que publica conteúdos voltados para a tecnologia de informação. A compatibilidade com os sistemas operativos escolhidos também se mostrou de extrema importância para a seleção dos navegadores. Assim, segue a lista dos que foram elegidos:

- *Chrome:* desenvolvido pela Google, é o principal navegador utilizado na Internet e integrado com todos os recursos oferecidos pelas plataformas de seus desenvolvedores.
- *Firefox:* navegador gratuito desenvolvido para ser leve, seguro, intuitivo, extensível e multiplataforma.
- *Opera:* presente no mercado há mais de 25 anos e que oferece aos utilizadores mais velocidade nos acessos e sistemas de proteção contra vulnerabilidades.

4.1.2 Ferramentas de Captura e Teste

Como descrito na seção 2.7.3, existe uma grande diversidade de ferramentas de análise e classificação do tráfego. Porém, devido às limitações ou à demanda por funcionalidades específicas, torna-se necessária a combinação dessas ferramentas para que se obtenham informações mais completas sobre os dados coletados.

A combinação de determinadas ferramentas tem como finalidade analisar os protocolos da camada de transporte, para identificação de fluxos TCP, fluxos UDP a utilizar QUIC e protocolos da camada de aplicação para identificação de sessões que utilizam protocolos de comunicação segura para os dados. É importante mencionar que não se busca identificar o tráfego encriptado e sim as informações encontradas antes desse processo. As seguintes ferramentas foram selecionadas:

Tcpdump

Utilitário de linha de comando que permite capturar e analisar o tráfego de rede que passa pelo sistema. Costuma ser utilizado no auxílio à identificação de problemas de rede, além de também ser uma ferramenta de segurança. Por ser uma ferramenta de linha de comando poderosa, versátil e com muitas opções de filtro, é ideal para executar em servidores ou dispositivos remotos para

[5]Os artigos publicados na revista encontram-se disponíveis em https://www.computerworld.com.

os quais uma interface gráfica não está disponível e para coletar dados que podem ser analisados posteriormente.

Para a coleta dos dados ser realizada, é necessário que os dois sistemas operativos estejam com a ferramenta *tcpdump* instalada corretamente. É importante ressaltar que essa aplicação precisa ser instalada com privilégios de administração do sistema para fornecer acesso a todas as suas funcionalidades oferecidas no processo de captura.

Tshark

Programa que analisa protocolos de rede, possibilita a captura ativa de pacotes e permite inspeção de todas as camadas de pilha protocolar TCP/IP. É um *sniffer* de rede desenvolvido para trabalhar através de linha de comandos. É compatível com diversos sistemas operativos, inclusive dispositivos móveis, como *smartphones* e *tablets*. Possui flexibilidade para interação com diversas linguagens de programação e possibilita a criação de *frameworks*[6] direcionados às necessidades requeridas em uma análise de tráfego específica. Esse programa também é encontrado em versão com interface gráfica, a qual leva o nome de *Wireshark*. Ele disponibiliza várias ferramentas complementares como código de cores para os protocolos, visualização do fluxo completo de uma conexão e elaboração de gráficos estatísticos dos dados coletados.

Libprotoident

Analisador que utiliza o processo de inspeção leve de informações (*LPI - Lightweight Packet Inspection*), o qual tem a função de investigar os primeiros quatro *bytes* da carga útil de um pacote juntamente com as portas e endereços IP usados entre dois pontos de conexão distintos. É considerado como uma forma limitada do processo de DPI, mas é avaliado com uma ferramenta de alta precisão para análise *off-line* [41]. Além de utilizar poucos recursos computacionais, possui uma biblioteca de suporte capaz de identificar mais de 200 protocolos distintos que atuam na camada de aplicação e utilizam TCP/UDP como transporte. Ainda, possui compatibilidade para ser inserida em uma *framework*.

LibTrace

Biblioteca que busca detalhes de compactação, formatos de captura e cabeçalhos de protocolo de forma mais eficiente, além de trabalhar em conjunto com a ferramenta *Libprotident*. A sua principal característica é a decodificação de protocolos para a camada de transporte e nas camadas

[6]Junção de várias ferramentas para criação de uma aplicação genérica.

inferiores. Esse processo suporta, por exemplo, a verificação de cabeçalhos IPv6 (*Internet Protocol Version 6*), IPv4 (*Internet Protocol Version 4*), VLAN(*Virtual Local Area Network*), MPLS (*Multiprotocol Label Switching*), PPPoE (*Point-to-Point Protocol over Ethernet*), fragmentos IP, cabeçalhos incompletos e até cabeçalhos encapsulados [42].

Tstat

É um analisador de rede passivo com capacidade de reconstruir as características do tráfego da Internet nos níveis IP, TCP/UDP e aplicação por intermédio de DPI [43]. Essa ferramenta disponibiliza diversas funcionalidades de pós-processamento que permitem a análise completa dos fluxos de dados transmitidos na rede. O Tstat possibilita retirar informações como: tamanho da janela de congestão, segmentos fora de sequência, segmentos duplicados, distinção entre cliente e servidores, aplicações específicas, protocolos, cabeçalhos, entre outras opções.

Fiddler

Desenvolvido para depuração de tráfego dos protocolos HTTP e HTTPS através de um servidor *proxy*. A ferramenta apresenta recursos para análise e inspeção de pedidos Web em traces já coletados ou em processos de captura em tempo real. O *Fiddler* pode ser usado com a maioria dos aplicativos sem a necessidade de configuração adicional, pois ao capturar ou analisar o tráfego, ele se registra no componente de rede da Internet do sistema operativo e solicita que todos os aplicativos comecem a direcionar seus pedidos para serem processados [44]. Essa ferramenta será complementar ao Tstat.

SSLAnalyzer

Funcionalidade da ferramenta *PcapPlusPlus*, que é uma biblioteca responsável por possuir recursos avançados para análise detalhadas de protocolos e camadas de um pacote [45]. O *SSLAnalyzer*, por sua vez, tem como principal característica a análise detalhada do tráfego SSL/TLS em traces coletados e em capturas em tempo real. A ferramenta também será complementar ao Tstat.

Robot Framework

É uma ferramenta gratuita para automação de testes através de processos automáticos. Como é uma aplicação aberta, permite criar soluções de testes automatizadas de forma elementar e

bastante flexível. O *framework* possui sintaxe de implementação simples pois utiliza comandos de fácil manipulação. Também possui recursos estendidos por bibliotecas de diversas linguagens de programação [46].

Uma das linguagens compatíveis é o *Python*, que disponibiliza uma biblioteca chamada *Selenium*. Essa biblioteca, por sua vez, é uma estrutura portátil para testar aplicações Web e fornece uma ferramenta de reprodução para criação de testes funcionais sem a necessidade de aprender uma linguagem de *script* específica para testes [47].

As ferramentas citadas, são totalmente compatíveis com os sistemas operativos escolhidos, mas necessitam estar instaladas nas máquinas virtuais juntamente com as bibliotecas que permitem a manipulação da linguagem *Python* e um programa para escrita do código. Apenas dessa forma as ferramentas podem ser utilizadas de forma correta e operacional.

4.1.3 Preparação dos Testes

Inicialmente uma estrutura de acesso foi arquitetada de modo a reproduzir de forma fiel um acesso a Internet por um utilizador em seu computador pessoal. Alguns trabalhos, como [34] e [37], visam mapear o comportamento de utilização da Web na ótica do utilizador e serviram de base para criar a ordem de acesso ao conteúdo proposto na Sessão 4.1.1. A sequência de acesso as URLs pré-definidas representa a melhor forma de capturar as informações, visto que se baseia apenas em volume de conteúdo acessado e não está associado a nenhum grupo de utilizadores com características específicos. O *script* que emulou o processo de navegação tinha como finalidade simular:

- abertura do navegador;
- inserção de URLs no navegador;
- navegação pelo endereço eletrónico inserido;
- execução de *streaming* de áudio e vídeo;
- navegação em um serviço de e-mail com envio e recebimento de mensagens;
- acesso e navegação em uma rede social onde fosse possível realizar interações com o conteúdo oferecido pela mesma;
- busca de termos específicos em vários motores de busca;
- fecho do navegador.

Baseado nos itens acima, um diagrama foi criado para elucidar a sequência de acessos no processo de emulação dos testes, conforme se pode verificar na Figura 4.1. Assim, é possível identificar a organização final do processo de simulação com a inserção dos endereços definidos.

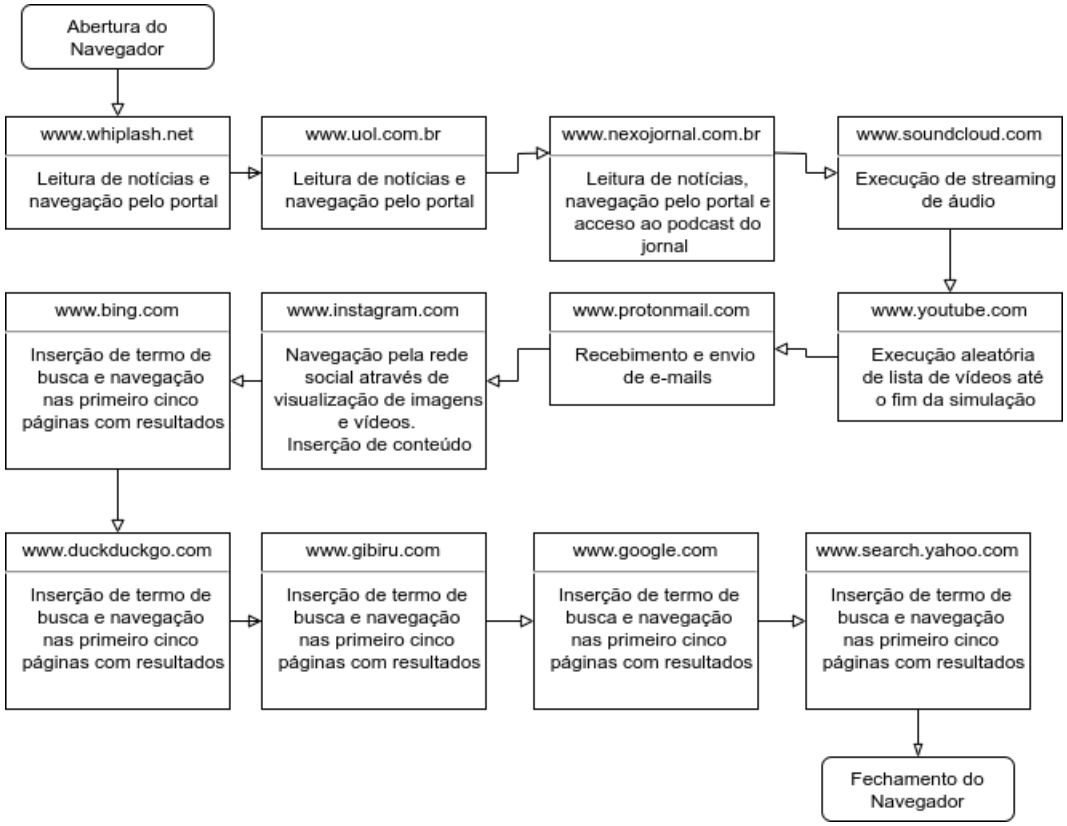

Figura 4.1: Processo de automação dos testes.

Em mais detalhe, primeiro ocorre o processo de abertura do navegador, depois a inserção e navegação nas URLs referentes a notícias: Whiplash, Uol e Nexo. No último endereço mencionado, é executado um *podcast*. Em um segundo momento ocorre a abertura do Youtube, que permanece aberto até o fim da simulação a executar vídeos de forma aleatória. Em paralelo às atividades de *streaming* de vídeo, decorre o acesso ao serviço de e-mail Protonmail com recebimento e envio de mensagens. Esse processo é sucedido com o acesso a rede social Instagram onde se realizou integrações com os serviços disponibilizados, que vão de visualização de fotos e vídeos à inserção de conteúdo por parte do utilizador. Por último ocorre o processo de busca de um termo aleatório em cada um dos motores de busca selecionados: Bing, DuckduckGo, Gibiru, Google e Yahoo.

A escrita do *script* foi realizada no *Pycharm*, que é um ambiente de desenvolvimento integrado (*IDE - Integrated Development Environment*) e fornece análise de código, depuração de erros,

testador de unidades, integração com sistemas de controle de versão e suporta o desenvolvimento de ferramentas Web [48]. Com o auxílio dessa ferramenta foi possível estruturar o *script* com a biblioteca *Selenium* e com o *framework* Robot. A estrutura do *script* é simples e possui quatro níveis de funções:

- *settings:* declaração das bibliotecas de teste utilizadas, neste caso a *Selenium*;
- *variables:* declaração das URLs e de listas de termos para utilização nos motores de busca;
- *test cases:* sequência de eventos realizados no navegador durante o processo de simulação;
- *keywords:* detalhamento das ações executadas pelo navegador através das funções do *framework*.

Na Sessão de *Scripts*, no Apêndice B, é possível visualizar um exemplo do código utilizado para realizar todo o processo de simulação de navegação. Ainda, toda vez que o *script* for executado ele sempre irá executar o navegador com as configurações padrões que são pré-estabelecidas em seu processo de instalação e que são configuradas pelas empresas que os desenvolvem.

4.1.4 Programação dos Testes

Após o término do programa da navegação simulada, o próximo passo consistiu em construir um *script* de rotina para otimizar o processo de testes e capturar todo o tráfego gerado. O *script* proposto necessitava ser genérico para ser compatível com os dois sistemas operativos propostos anteriormente, além de poder ser executado em linha de comando ou através de ícone de atalho em modo gráfico do sistema operativo. Assim, as seguintes operações foram adotadas:

1- mapeamento da data e hora que o arquivo está a ser gerado;

2- inserção do caminho onde o arquivo que contém o tráfego gerado será armazenado;

3- inicialização do utilitário que fará a captura dos dados;

4- inicialização da rotina de testes com os *scripts* gerados pelo Robot *framework*;

5- o arquivo gerado é armazenado com a data e hora da execução do teste;

6- a ferramenta de captura é finalizada.

Como foram utilizado três navegadores, a rotina contou com uma etapa específica de 45 minutos em média para cada um. Esse detalhe mostrou-se importante para gerar um arquivo independente com o tráfego gerado por cada navegador, pois facilitou a etapa de tratamento de todos os dados coletados. A utilização de dois sistemas operativos distintos necessitou de breves modificações com relação à hora e data do sistema, mas para todos os outros parâmetros praticamente não houve modificações. Na seção B1 e B2, do Apêndice B, é possível verificar os *scripts* utilizados em cada sistema operativo e suas respectivas diferenças.

4.1.5 Processo de Medição

A captura dos dados foi realizada em um ambiente residencial, através de simulação de um computador pessoal virtualizado, com acesso à Internet através de banda larga dedicada durante um período de 7 dias. A Figura 4.2 ilustra de forma sucinta, o ambiente utilizado onde é possível verificar um terminal com as ferramentas de captura e as máquinas virtuais instaladas. Também é representado em detalhes quais ferramentas estavam instaladas, em cada estação, de forma a ilustrar o processo de teste e recolha dos dados.

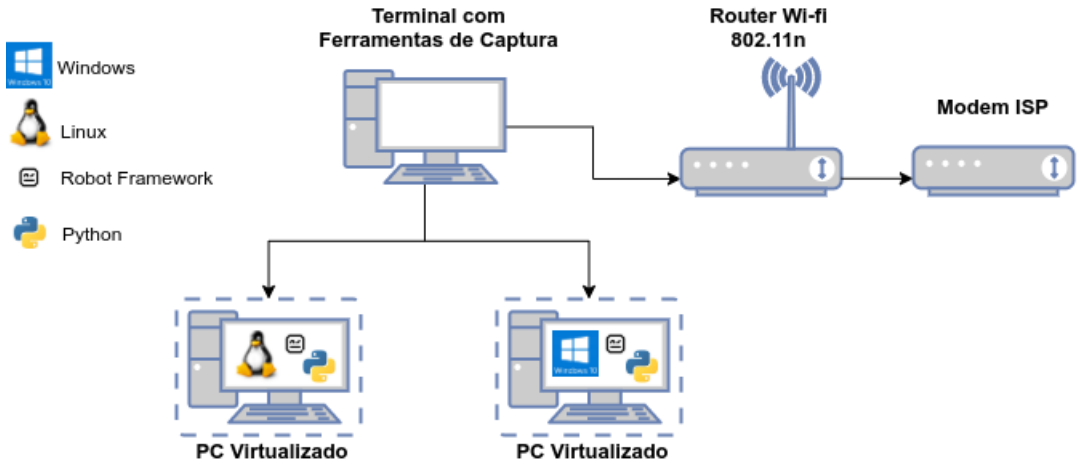

Figura 4.2: Topologia de testes.

O processo de simulação para coleta dos dados ocorre conforme as seguintes etapas:

1- inicialização do computador hospedeiro das máquinas virtuais;

2- inicialização da primeira máquina virtual, com o sistema operativo Xubuntu;

3- inicialização do *Script* de rotina com e execução de uma sub-rotina para emulação de navegação;

4- recolhimento do tráfego gerado em arquivos separados por data e hora;

5- finalização da primeira máquina virtual;

6- inicialização da segunda máquina virtual, com o sistema operativo Windows;

7- inicialização do*Script* de rotina com e execução de uma sub-rotina para emulação de navegação;

8- recolhimento do tráfego gerado em arquivos separados por data e hora;

9- finalização da segunda máquina virtual.

4.1.6 Características dos Navegadores

Os navegadores escolhidos possuem características distintas e com o principal objetivo de oferecer a melhor alternativa de navegação para seus utilizadores. As alternativas são refletidas em melhorias de desempenho, segurança, privacidade e diversos outros atributos que podem ser consultados diretamente nas URLs do Google[7], Firefox[8] e Opera[9].

Em questões de desempenho, de acordo com [49], os navegadores costumam ter recursos de otimização fundamentados em quatro pilares:

- *pré-busca e priorização de recursos:* análises de pedidos podem oferecer informações extras à camada de rede para indicar a prioridade relativa de cada recurso e para otimizar desempenho. Assim, os pedidos de alta prioridade são logo processados com todos os recursos disponíveis. Já os pedidos de baixa prioridade podem ser retidos temporariamente em uma fila de espera. Essa função busca diminuir a latência, preferência de conteúdo do *payload*, entre outros;

- *pré-resolução de DNS:* o nome de servidores são pré-resolvidos com antecedência para evitar atrasos nas buscas de DNS perante solicitações HTTP/HTTPS. Esse tipo de recurso pode ser ativado através do histórico de navegação armazenado, ação dos utilizadores ou outros identificadores na página Web;

- *pré-conexão TCP:* após uma resolução de DNS, o navegador especula a conexão TCP para antecipar uma conexão HTTP de forma a eliminar outro TCP handshake entre o cliente e servidor. Assim, busca diminuir a latência da rede;

[7]Pode ser consultado em https://www.google.com/intl/pt-PT/chrome.
[8]Pode ser consultado em http://br.mozdev.org.
[9]Pode ser consultado em https://www.opera.com/pt.

- *pré-renderização da página:* alguns navegadores permitem recomendar o próximo destino provável de conexão e podem pré-renderizar uma página Web inteira ocultamente, para ser trocada de forma instantânea quando o utilizador iniciar a navegação.

Os navegadores escolhidos, baseados nos itens acima, podem ter suas funcionalidades exploradas de forma positiva com a melhoria de desempenho entre informações trocadas entre cliente-servidor. Porém, essas funções também permitem alguns programas se aproveitarem de vulnerabilidades já conhecidas para realizar os mais diversos tipo de ataques [8].

Em termos de recursos de segurança, são oferecidos serviços em que o utilizador tem garantia da encriptação de dados e conexões, navegação anónima e a não execução automática de *scripts* possivelmente maliciosos. Os navegadores também buscam proteger os utilizadores contra fraudes mediante o bloqueio de URLs considerados como potenciais ameaças. Empresas especializadas em segurança na Internet realizam a criação e divulgação de listas com URLs nocivas à navegação do utilizador. Outro recurso de segurança dos navegadores é de bloquear anúncios que também podem ser nocivos à integridade dos dados do utilizador

Todos os navegadores também são considerados multiplataforma, pois têm compatibilidade com uma grande quantidade de dispositivos e diversos sistemas operativos. Desta forma, precisam ter recursos personalizáveis que permitam a adição dos mais diversos tipos de programas e serviços ao navegador. Todos os três navegadores selecionados possuem uma imensa biblioteca de extensões com o objetivo de melhorar a integração de funções e o processo de navegação.

A compatibilidade com as versões do protocolo HTTP é existente em todos os navegadores paras versões 1.1 e 2.0. A versão 3.0, por ser experimental, encontra-se ainda indisponível para utilização.

Um ponto de extrema importância é a política de privacidade, que consiste em termos que vão garantir a segurança e a real utilização dos dados dos utilizadores. Cada navegador tem a obrigação de disponibilizar, de forma acessível, esclarecimentos sobre compartilhamento dos dados, dúvidas referente a cadastro, uso de *cookies* e informações que evidenciem o envolvimento e compromisso com seus clientes. Desta forma, os termos representam um contrato firmado entre o utilizador e o navegador. Ao aceitar esse contrato, o utilizador fica em acordo com todas as regras propostas no documento. Na Tabela 4.1 é possível conferir de forma resumida alguns termos presentes na política de privacidade proposta pelos navegadores: Google[10], Firefox[11] e Opera[12].

[10]Pode ser consultado em https://policies.google.com/privacy.
[11]Pode ser consultado em https://www.mozilla.org/pt-BR/privacy/firefox/.
[12]Pode ser consultado em https://www.opera.com/pt/privacy.

Tabela 4.1: Termos da política de privacidade dos navegadores.

Termos aceites na política de privacidade	Google	Firefox	Opera
Coleta de informações do sistema operativo	Sim	Sim	Sim
Coleta de informações pessoais e senhas	Sim	Sim	Sim
Coleta de pesquisas realizadas	Sim	Sim	Sim
Coleta de informações de voz e áudio	Sim	Sim	Sim
Coleta das atividades de compras	Sim	Sim	Sim
Coleta de contatos comunicados e conteúdos compartilhados	Sim	Sim	Sim
Coleta das atividades em sites de terceiros	Sim	Sim	Sim
Coleta do histórico de navegação	Sim	Sim	Sim
Coleta de cookies	Sim	Sim	Sim
Coleta das informações de localização geográfica	Sim	Sim	Sim
Coleta do endereço IP	Sim	Sim	Sim
Dados de pontos de acesso Wi-Fi, torres de celular e Bluetooth	Sim	Não	Não
Estatísticas de uso e relatórios de erros	Sim	Sim	Sim
Compartilhamento de dados com terceiros	Sim	Sim	Sim
Modificação dos dados compartilhados pelo utilizador	Sim	Sim	Sim
Declaração de que os dados compartilhados são anonimizados	Não	Não	Sim
Informação do período de retenção dos dados coletados	Sim	Não	Não

4.1.7 Características dos Motores de Busca

Um motor de busca é projetado para oferecer a seus utilizadores um programa que localize informações armazenadas em um sistema computacional baseado em palavras-chaves. Com essas características, cada motor de busca entrega resultados relevantes, opções de ampliação e restrições de pesquisa, organização das informações encontradas e fácil compreensão dos resultados. Outra propriedade relevante é a compatibilidade com o protocolo HTTPS, que representa a capacidade de utilizar criptografia para proteção de conexões cliente/servidor e o redirecionamento para conteúdos seguros.

Ao realizar uma pesquisa, o utilizador necessita de uma rápida resposta do motor de busca. Para que a o processo seja eficiente e retorne informações relevantes, um motor de busca requer a execução prévia de algumas funções específicas:

- *coleta:* programas que percorrem continuamente uma vasta quantidade possível de páginas Web;
- *armazenamento:* as páginas encontrada são armazenadas em servidores dedicados;
- *extração:* todos os conteúdos das páginas encontradas são extraídos;

- *indexação:* o conteúdo extraído precisa ser estruturado em meios eficazes de pesquisa;
- *classificação:* método para colocar páginas Web com mais probabilidade de dar a resposta correta para a palavra solicitada durante o processo de busca;
- *pesquisa:* forma de como o motor de busca apresenta os resultados encontrados que podem se basear, ou não, em critérios opcionais impostos pelo utilizador.

As funções descritas acima destinam-se a clarificar, de forma breve, a capacidade do motor de busca conseguir processar, pesquisar e entregar resultados de forma eficiente para o utilizador. Através de algorítimos de processamento, muitos motores de busca garantem esses resultados e empenham-se em apresentar conteúdos totalmente seguros com o redirecionamento para URLs que utilizem o protocolo HTTPS como padrão.

O emprego da política de privacidade também acontece nos motores de busca e de forma semelhante ao que é empregue nos navegadores. A Tabela 4.2 resume alguns termos empregues pelos motores Google, Bing[13], Yahoo[14], DuckduckGo[15] e Gibiru[16].

Tabela 4.2: Termos da política de privacidade dos motores de busca.

Características	**Google**	**Bing**	**Yahoo**	**DuckDuckGo**	**Gibiru**
Coleta de informações pessoais	Sim	Sim	Sim	Não	Não
Coleta de pesquisas realizadas	Sim	Sim	Sim	Não	Não
Coleta do histórico de pesquisa	Sim	Sim	Sim	Não	Sim
Resultados personalizados	Sim	Sim	Sim	Não	Não
Estatísticas de utilização	Sim	Sim	Sim	Não	Não
Compartilhamento de dados com terceiros	Sim	Sim	Sim	Não	Não
Coleta e inserção de cookies*	Sim	Sim	Sim	Não	Não
Utilização de busca patrocinada**	Sim	Sim	Sim	Não	Não
Detector de políticas de privacidade abusiva***	Não	Não	Não	Sim	Não
Coleta da localização geográfica	Sim	Sim	Sim	Sim	Sim
Coleta de endereço IP	Sim	Sim	Sim	Não	Não
Navegação segura com HTTPS	Sim	Sim	Sim	Sim	Sim
Período de retenção dos dados coletados	Sim	Sim	Não	Não	Não

(*) Além da coleta de cookies utilizados, o motor de busca também os implanta para acesso a informações do computador utilizado.
(**) Processo onde as páginas Web pagam aos motores de busca para exibirem seu endereço em pesquisas relevantes.

[13]Pode ser consultado em https://privacy.microsoft.com/pt-pt/privacystatement
[14]Pode ser consultado em https://policies/verizonmedia/privacy/products/searchservices/pt/br/index.html
[15]Pode ser consultado em https://duckduckgo.com/privacy
[16]Pode ser consultado em https://gibiru.com/privacy-policy/privacy-policy

(***) Localização de páginas Web com rastreadores ocultos e políticas de privacidade que expõem os dados do utilizador.

4.1.8 Reprodutibilidade

A reprodução do ambiente de testes torna-se possível para quaisquer *scripts*, rotinas de testes e capturas de tráfego efetuadas neste trabalho. Para que seja viável, torna-se essencial respeitar as ferramentas citadas ao longo de toda Sessão 4.1, bem como suas versões mais atualizadas para garantia de um bom desempenho. Para este trabalho, as seguintes versões foram utilizadas:

- *Virtual Box* na versão 6.1.6;
- Distribuição Linux Xubuntu na versão 18.04 com todas atualizações vigentes até maio de 2020;
- Windows na versão 10 com todas as atualizações vigentes até maio de 2020;
- Linguagem de programação *Python* na versão 3.7;
- Para escrita de código o *PyCharm* foi utilizado na versão 2020.1;
- *Robot Framework* na versão 3.1.2;
- *Selenium Library* na versão 4.4.0;
- *Tshark* na versão 2.6.10;
- *Tcpdump* na versão 4.9.3;
- *Fiddler* na versão 0.5.0;
- *SSLAnalyzer* na versão 19.12;
- *Google Chrome* na versão 81;
- *Firefox* na versão 75;
- *Opera* na versão 68.

4.2 Extração das Informações

Para realizar o processo de extração dos dados é necessário descrever o papel de cada ferramenta apresentada na Sessão 4.1.2. Com a junção de toda a informação se pretende ter mais precisão na análise dos dados. As funcionalidades exploradas em cada ferramenta são a seguir descritas.

Tcpdump

Responsável por executar a captura do tráfego durante o período de testes. Os arquivos gerados serão salvos na extensão *.pcap*, visto que todas as ferramentas selecionadas possuem compatibilidade de leitura.

Tshark

Utilizado para tratamento dos dados capturados devido à sua capacidade de inserção de filtros, verificação de fluxos e geração de gráficos estatísticos.

Libprotoident

O *Libprotoident* necessita da biblioteca *Libtrace* instalada para ativar suas funções e ter a eficiência esperada. Possui duas funções a *lpi_protoident* e *lpi_find_unknown*. A função *lpi_protoident* é responsável por tentar identificar os fluxos individuais dos dados que foram concluídos ou expirados e expor informações importantes da camada de aplicação. Os campos relevantes dessa função são:

Protocolo de aplicação utilizado

IP cliente
IP servidor
Porta usada pelo cliente
Porta usada pelo servidor
Protocolo de transporte (6 = TCP, 17 = UDP)
Registro de data e hora de início do fluxo
Registro de data e hora do fim do fluxo
Total de bytes enviados do cliente para o servidor
Total de bytes enviados do servidor para o cliente
Primeiros quatro bytes de carga útil enviados do cliente em hexadecimal
Primeiros quatro bytes de carga útil enviados do cliente em ASCII
Tamanho do primeiro pacote de carga útil enviado do cliente
Primeiros quatro bytes de carga útil enviados do servidor em hexadecimal
Primeiros quatro bytes de carga útil enviados do servidor em ASCII
Tamanho do primeiro pacote de carga útil enviado do servidor

A segunda, *lpi_find_unknown*, é responsável por listar os fluxos que a função *lpi_protoident* não conseguiu identificar. Possui os mesmos campos da primeira função, com exceção do campo de protocolo da aplicação.

Fiddler

Possibilita visualizar as informações de cada objeto de sessão detalhada com requisição/respostas da conexão entre cliente-servidor. O objeto de sessão também mantém um conjunto de sinalizadores que registram metadados sobre a sessão e um temporizador que armazena carimbos de data e hora registrados no decorrer do processamento da sessão.

SSLAnalyzer

Realiza detalhamento exclusivo do tráfego SSL/TLS e apresenta as seguintes informações:

Contagem e taxa de pacotes

Largura de banda
Contagem e taxa de fluxo
Pacotes e dados médios por fluxo
Número de mensagens do cliente-hello e do servidor-hello
Número de fluxos SSL com handshake e mensagens de alerta bem-sucedidas
Histograma do nome do host
Histograma do conjunto de cifras
Histograma de versão Hello do cliente
Histograma de portas SSL / TLS

Tstat

De todas as ferramentas, o *TSTAT* terá importância maior devido a ter a propriedade de relatar todas as conexões TCP e cabeçalhos HTTP/HTTPS mais detalhadamente. Para validar a conexão, a aplicação identifica o primeiro segmento de início de sessão SYN ( ) e seu término quando são observados os segmentos FYN ( ), ACK ( ) e RST ( ). Para toda a conexão TCP fechada corretamente, a ferramenta cria um arquivo de eventos (*log*) chamado log_tcp_complete e para conexões incompletas, cria outro arquivo chamado log_tcp_nocomplete.

Os registros contidos nas saídas de *log* são configuráveis a partir de um arquivo chamado *runtime.conf*. Esse arquivo é separado em módulos para determinar quais serão as informações analisadas. A princípio, três funcionalidades estarão ativas: *Core TCP set*, encarregado de ativar os eventos relativos às conexões TCP concretizadas, *TCP Layer7 set*, que inclui informações da camada de aplicação, e a função *log_ http_ complete* que registra todas as requisições e respostas do protocolo HTTP/HTTPS em um arquivo específico.

As funções *Core TCP set* e *TCP Layer7 set* são registradas no log_tcp_complete e todas as suas saídas podem ser consultadas no Apêndice A. Os parâmetros escolhidos para análise foram:

- *connection type:* registro a indicar o tipo de conexão (HTTP, SSL/TLS) conforme identificado pelo mecanismo de inspeção de camada de aplicação;
- *HTTP type:* estados do protocolo HTTP;
- *HTTP request count:* número de requisições para conexões HTTP entre cliente/servidor;
- *HTTP response count:* número de requisições para conexões HTTP entre servidor/cliente;
- *first HTTP response:* código de resposta HTTP;
- *PSH-separated C2S:* mensagens de notificação entre cliente/servidor;

- *PSH-separated S2C:* mensagens de notificação entre servidor/cliente;
- *TLS client hello SNI:* nome do servidor indicado pelo cliente;
- *TLS client NPN/ALPN:* negociação SSL/TLS para HTTPS entre cliente/servidor;
- *TLS server NPN/ALPN:* negociação SSL/TLS para HTTPS entre servidor/cliente;
- *TLS client ID reuse:* número de identificação de sessão utilizado pelo cliente;
- *TLS client last handshake:* último *handshake* realizado pelo cliente antes do envio dos dados encriptados;
- *TLS server last handshake:* último *handshake* realizado pelo servidor antes do envio dos dados encriptados;
- *TLS client app data time:* tempo entre a primeira mensagem de dados e o primeiro fluxo enviado do cliente para o servidor em milissegundos;
- *TLS server app data time:* tempo entre a primeira mensagem de dados e o primeiro fluxo enviado do servidor para o cliente em milissegundos;
- *TLS client app data bytes:* sequência de mensagens entre cliente/servidor;
- *TLS server app data bytes:* sequência de mensagens entre servidor/cliente;
- *FQDN:* nome do servidor de domínio recuperado;
- *IP of DNS resolver:* endereço do servidor DNS utilizado;
- *DNS request time:* tempo de requisição ao DNS em milissegundos;
- *DNS response time:* tempo de resposta do DNS em milissegundos.

A função *log_http_complete* registra as informações de cabeçalho de requisição e resposta do protocolo HTTP/HTTPS. Os parâmetros selecionados para análise são:

- *client IP addr:* endereço IP do cliente;
- *client TCP port:* porta TCP do cliente;
- *server IP addr:* endereço IP do servidor;
- *server TCP port:* porta TCP do servidor;
- *segment time abs:* tempo absoluto do processo de requisição e resposta;

- *request method:* método de requisição (GET/POST/HEAD);
- *hostname:* nome de domínio do servidor (para hospedagem virtual) e (opcionalmente) o número da porta TCP na qual o servidor está a escutar;
- *FQDN:* nome do servidor de domínio recuperado;
- *URL path:* URL de requisição
- *referer:* origem da solicitação do cliente;
- *user agent:* identificador do navegador do cliente;
- *cookie:* cabeçalho de solicitação de *cookie* recebido pelo navegador do cliente;
- *do not track:* opção de não rastreio habilitada pelo navegador do cliente;
- *response string:* identificador de resposta do servidor;
- *response code:* código de resposta enviado pelo servidor;
- *content len:* tamanho do corpo da entidade em *bytes*;
- *content type:* tipos de arquivo contidos no corpo da entidade;
- *server:* mensagem de resposta do servidor;
- *range:* resposta HTTP/HTTPS para intervalos de solicitações parciais;
- *location:* registro de redirecionamento de URL;
- *set cookie:* envio de *cookies* do servidor para o cliente.

O fluxograma presente na Figura 4.3 ilustra como será o processo de recolha das informações a partir de cada ferramenta descritas na Sessão 4.1.2.

4.2.1 Tratamento e Classificação dos Dados

Com as informações extraídas, são identificados quais os dados relevantes do utilizador que ficam disponíveis e/ou expostos em um processo de captura. Dessa forma, é possível constatar até que ponto sua navegação pode ser dita como segura. Para isso ser factível, as ferramentas selecionadas são essenciais para averiguação dos dados a serem capturados precisam evidenciar, se possível:

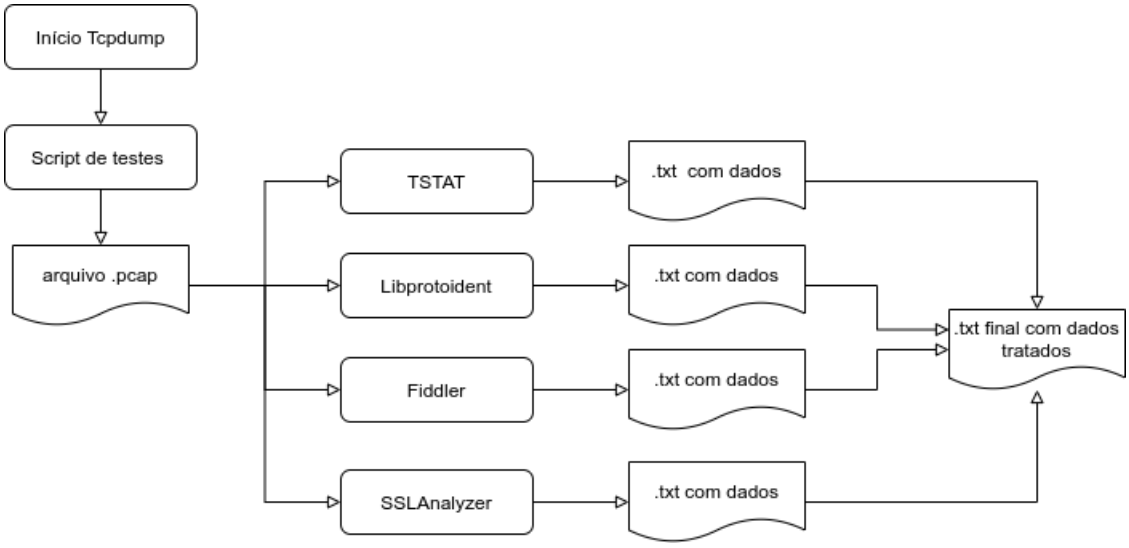

Figura 4.3: Fluxograma de utilização das ferramentas.

- sessões de *handshake* de três vias que antecedem a encriptação do *payload*;
- informações de segurança relativas a versão dos protocolos HTTP, HTTPS, TLS e SSL;
- informações de cabeçalho de camada de transporte e aplicação que são visíveis;
- informações de *cookies*;
- informação de certificados;
- processo de redirecionamento para conteúdo seguro do protocolo HTTPS;
- tempo médio entre requisições e respostas entre cliente/servidor.

Todos os dados recolhidos devem passar por um processo de tratamento em etapas, de modo que se extraia as informações mais relevantes de todo o processo de utilização do protocolo HTTPS e HTTP. A Figura 4.4 demonstra o processo de extração do fluxo de dados aplicado em todas as conexões TCP realizadas durante o todo período de medição. A partir das conexões TCP completas, um *script* de filtragem de dados irá separar os fluxos por conexões seguras (HTTPS) e conexões inseguras (HTTP). Dos dois tipos de conexões, se extrairá todas as informações do processo completo de comunicação entre cliente/servidor. Apenas após esse processo os dados são, de facto, interpretados. Para as conexões TCP incompletas e conexões TCP completas, com informações do processo de comunicação cliente/servidor parciais, os dados não terão validade para o propósito do trabalho e devem ser descartados.

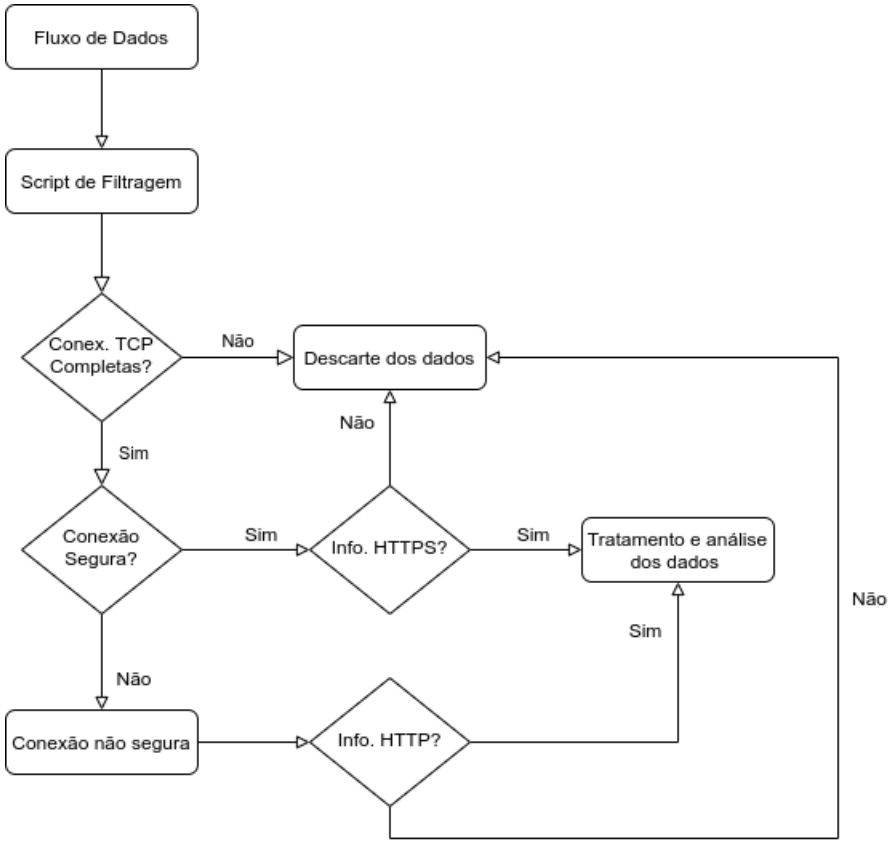

Figura 4.4: Fluxograma de tratamento dos dados.

4.3 Sumário

Este capítulo mencionou a metodologia utilizada para a criação de um ambiente de testes funcional e das ferramentas escolhidas, com destaque para as funcionalidades que serão aproveitadas, para realizar o processo de captura e análise dos dados. Também foi o processo das sessões de teste e os períodos de medição. Por último foi exposto em detalhe quais as informações pertinentes de cada ferramenta que foram extraídas para uma análise detalhada de todos os dados colhidos. Todas essas informações são necessárias, pois são alvo de análise no capítulo seguinte.

Capítulo 5

Cenários de Testes e Resultados

Este capítulo se propõe a descrever toda a análise dos dados encontrados no processo de medição realizado no ambiente experimental. E é dividido em duas partes. A primeira parte enfatiza a objetividade dos testes juntamente com uma taxonomia de todos os processos para investigação dos dados. A segunda e maior parte deste capítulo traz a análise de todos os dados levantados no processo de simulação de um acesso a Internet por um utilizador final, com o intuito de verificar alguns aspectos da implantação de comunicação segura com o protocolo HTTPS.

5.1 Objetivo dos Testes

O principal propósito dos testes é realizar um estudo do comportamento do protocolo HTTPS empregado no acesso à Internet por um utilizador comum. Primeiramente pretende-se capturar o tráfego produzido no ambiente de testes de acordo com as premissas estabelecidas na Sessão 4.2. Esse tráfego armazenado, por sua vez, representa as informações capturadas no processo da navegação em um grupo definido de URLs e pesquisas realizadas em motores de busca. Em um segundo momento as informações são submetidas a métodos avaliativos para identificar quais tipos de impactos a navegação segura proporciona ao utilizador. Por último busca-se investigar quais informações, de facto, ficam expostas durante a troca de informações entre as máquinas cliente/servidor.

Os resultados apresentados nas sessões seguintes foram validados por testes preliminares executados em um período de quatro dias. Esses testes buscaram confirmar a eficiência do processo de simulação da navegação e verificar se as saídas geradas possuíam ou não, informações significativas para serem analisadas. É possível verificar, através da Figura 5.1, que o tempo de navegação ficou

equivalente nos dois sistemas operativos. Isso demonstra que a combinação das ferramentas de teste utilizadas, (*selenium + robot framework*) possuem o mesmo comportamento independente do ambiente no qual estão a ser executadas.

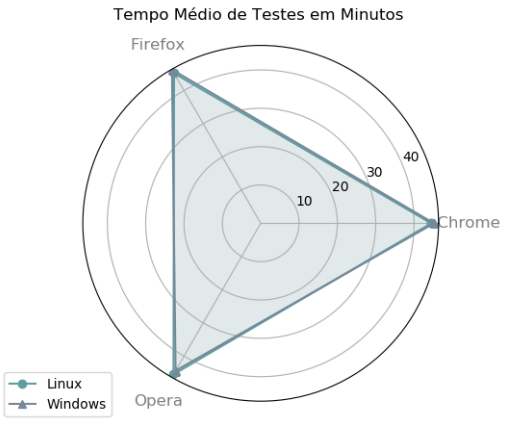

Figura 5.1: Tempo médio dos traces realizados por navegador.

Como se pode observar, cada etapa de teste de um navegador durou em média 45 minutos. Como foram utilizados dois sistemas operativos, contabilizou-se o tempo de 135 minutos para cada um. Assim, cada ciclo diário de testes foi efetuado no período de 270 minutos que correspondeu à 4 horas e 30 minutos, somando mais de 9 horas de capturas de tráfego e 3,4 GB em informações úteis. Para os testes reais, realizados em um intervalo de sete dias, cada ciclo diário teve o mesmo período. Porém, somando mais de 30 horas de capturas de tráfego e 7,5 GB em informações.

5.1.1 Taxonomia dos Testes

Conforme a Sessão 4.1.3, que propõe uma sequência de navegação, foram planejados dois cenários de testes. O primeiro cenário representa o sistema operacional Windows como simulação da navegação realizada através dos navegadores Google, Firefox e Opera. O segundo cenário corresponde ao sistema operacional Linux, mas também a realizar a mesma simulação de navegação e a utilizar os mesmos navegadores. Na tabela 5.1 pode-se verificar que cada processo de simulação é dividido em seis etapas de investigação. Cada etapa consiste em procedimentos específicos na interação cliente/servidor que possam gerar as informações necessárias para análise. As etapas são:

- *conexões TCP:* acesso a todas URLs selecionados para dispor de visibilidade das conexões

TCP completas e informações da camada de rede;

- *conexões SSL/TLS:* acesso a URLs selecionados para registrar o processo *handshake* que antecede a encriptação do *payload*;

- *HTTP e HTTPS:* acesso a URLs selecionados para verificar encriptação do conteúdo acessado;

- *cookies:* acesso a URLs selecionadas para observar como os cookies interagem com o navegador do utilizador.

- *info_navegador:* acesso a URLs selecionados para registrar o comportamento dos navegadores no registro de informações da navegação do utilizador;

- *pesquisas:* processo de pesquisa de palavras comuns para registrar o comportamento dos motores de busca.

Após o processo de produção do tráfego, os dados gerados serão classificados em níveis de exposição de dados, vulnerabilidades e utilização de tráfego encriptado. Essa classificação tem o intuito de perceber o que a utilização crescente do HTTPS proporciona ao utilizador final.

Tabela 5.1: Taxonomia dos testes.

Cenários	**Ambiente**	**Navegador**	**Etapas de Estudo Para cada Cenário**	**Característica de Verificação em cada Etapa***	**Classificação****
Cenário 1	Windows	Chrome Firefox Opera	1- Conexões TCP 2- Conexões SSL/TLS 3- HTTP x HTTPS	ED/V ED/V V/TE	E ou I E ou I E ou I
Cenário 2	Linux	Chrome Firefox Opera	4- Cookies 5- Info_navegador 6- Pesquisas	ED/V ED/V/TE ED/V/TE	E ou I E ou I E ou I

(*) Para as características de verificação de cada etapa, serão adotados os seguintes critérios: ED - Exposição de dados, V - Vulnerabilidades e TE - Tráfego Encriptado.
(**) Para os níveis de classificação são adotados os critérios: E - Existente e I - Inexistente.

5.2 Análise dos Resultados

5.2.1 Conexões TCP

Primeiramente, buscou-se classificar o tráfego das conexões TCP completas provenientes de todas as sessões de *handshake* da comunicação entre cliente-servidor. A separação por sessões contribuiu para identificar quais informações ficam visíveis durante esse processo. A Figura 5.2 demonstra a quantidade de conexões TCP completas e não completas por cada sistema operacional. No Linux, Figura 5.2a, a quantidade de conexões efetivadas pelos três navegadores supera a média de 99% e para as conexões não completadas, menos de 1%, o que representa uma pequena instabilidade do sinal conexão Wi-Fi no ambiente de testes. Já a representação da Figura 5.2b retrata o comportamento dos navegadores no Windows, o qual é semelhante em 98.6% para conexões realizadas e 1.4% para não realizadas.

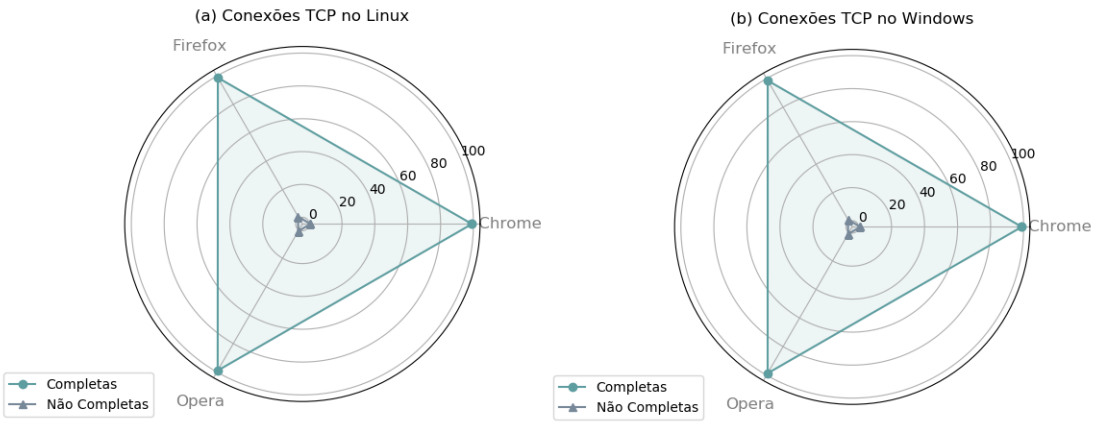

Figura 5.2: Percentual de conexões TCP no Linux e Windows.

A pequena diferença encontrada entre os sistemas, é relacionada à quantidade de pacotes TCP retransmitidos devido a erros por parte do roteador. No entanto, por se tratar de interferências comuns em uma residência no sinal Wi-Fi local, não ocasionou nenhum tipo de complicação aos processos de teste e medições. Essa comparação demonstra que os navegadores tiveram o comportamento bem similar quando se trata em estabelecer otimização de desempenho em realizar conexões TCP de forma eficiente. Os três navegadores utilizados, em concordância com a Sessão 4.2.7, procuram potencializar recursos para uma melhor experiência na utilização através de configurações para alta performance e para conexões TCP.

Outro ponto importante é a existência de uma predominância do protocolo IPv4 nas conexões TCP. A Figura 5.3 evidencia que nenhuma das URLs, mencionadas na Sessão 4.1.1, fazem utiliza-

ção do IPv6 no momento em que este trabalho foi desenvolvido. Segundo o endereço eletrônico de monitoramento da Akamai[1], a adoção do IPv6 está a ocorrer gradativamente ao redor do mundo. Isto é, por ser um processo que envolva mudanças de infraestrutura nos provedores de serviços e necessite de grandes investimentos, torna-se bastante lento. Para o utilizador final o IPv6, em questões de segurança, trás benefícios com a implantação do Protocolo de Segurança IP[2] (*IPSEC - IP Security Protocol*), por exemplo. Até a adoção total desse novo protocolo de rede, o utilizador final ainda continuará exposto as falhas e limitações conhecidas do IPv4 [8]. Porém, isto não significa que endereços IPv6 não estejam a ser utilizados pelos servidores Web e ISPs pelos quais o utilizador realiza seu acesso. Esse protocolo pode estar a ser utilizado, mas nesse caso específico não foi visível pelo lado do utilizador.

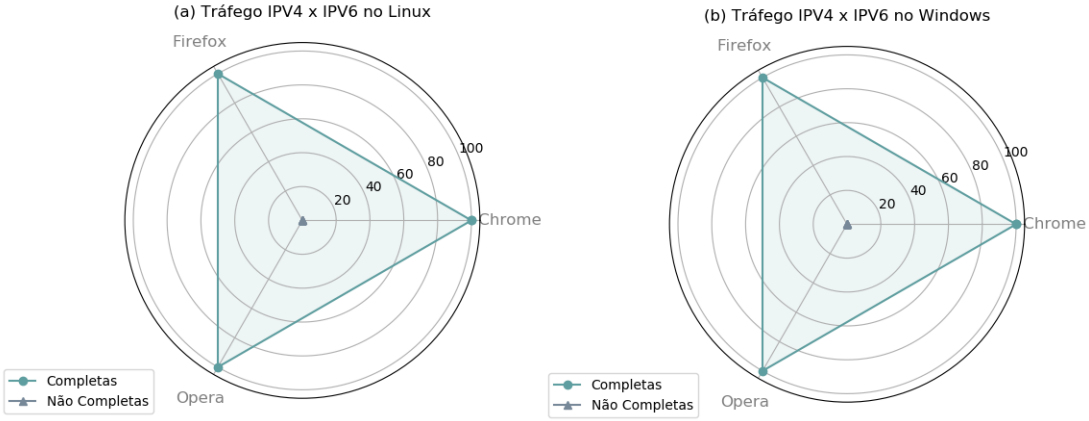

Figura 5.3: Percentual de Tráfego IPv4 e IPv6 nos sistemas operativos Linux e Windows.

A vulnerabilidade através de informações da camada de rede é um assunto bastante discutido, como referido em [38]. Mas uma particularidade que chamou bastante atenção foi a presença de pacotes do protocolo SSDP (*Simple Service Discovery Protocol*), que é um protocolo utilizado para descobrir outros dispositivos na rede na qual o protocolo esteja ativo. Neste caso, os navegadores Chrome e Opera por fazerem uso do mesmo motor de navegação[3], apresentam a maior quantidade de pacotes. Os pacotes encontrados são solicitações do navegador para localizar e sincronizar dispositivos conectados aos serviços digitais do Google. A Figura 5.4 sumariza essas informações e demonstra também que o Firefox não faz uso desse protocolo.

[1]Mais informações em https://www.akamai.com.

[2]Protocolo desenvolvido para suprir a falta de segurança de informações trafegando em rede pública. O IPSec realiza a proteção dos pacotes IP de dados privados, encapsulando em outros pacotes IP para serem transportados.

[3]Motor de navegação Blink, o qual é desenvolvido pela Google e possui a função de compilar páginas HTML em conteúdo visível pelo navegador.

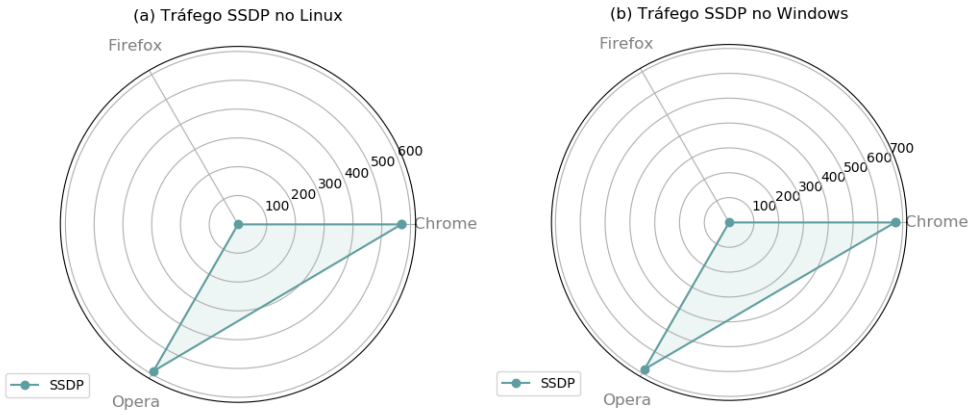

Figura 5.4: Quantidade de pacotes do protocolo SSDP presentes no Linux e Windows.

Mesmo sendo um recurso benéfico, já se tem conhecimento que a utilização do protocolo SSDP pode amplificar ataques DoS [8] [14]. Isso ocorre devido a uma grande quantidade de dispositivos na rede estarem aptos a executar esse protocolo e auxiliar um atacante a propagar melhor o ataque. Essa situação pode ser agravar porque o computador do utilizador está conectado a Internet. Por se tratar de uma configuração que vem por padrão nos navegadores, torna-se um grande problema na difusão de ataques.

As conexões TCP executadas pelos navegadores levantaram outra questão significativa: a exposição de portas de comunicação utilizadas pelo lado do cliente [16]. Por ser uma das técnicas mais antigas de caracterização do tráfego e com a evolução da tecnologia, as portas de serviços mais conhecidas passaram a ser substituídas por portas aleatórias de forma a evitar ataques. A divulgação dessas portas abertas por parte do navegador acabam por deixar os utilizadores vulneráveis a esses ataques. Na Figura 5.5 pode-se identificar que o navegador Chrome e Opera deixam visíveis uma quantidade significativa de portas utilizadas na conexão com os servidores Web. Já o Firefox suprime essas informações.

De forma resumida, a primeira etapa destaca algumas vulnerabilidades existentes da comunicação dos navegadores com a camada de rede. Como foi mostrado, os navegadores Chrome e Opera possuem mais aspectos de exposição a certas informações do utilizador que o Firefox. Muitas configurações são inseridas automaticamente no processo de instalação do navegador e o utilizador final, sem conhecimento de configurações, não é capaz de identificar tais fragilidades.

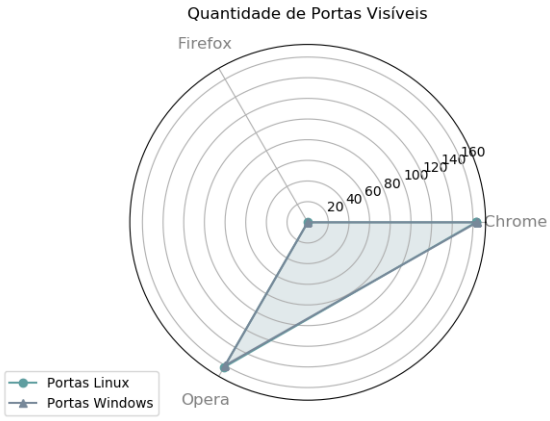

Figura 5.5: Quantidades de portas visíveis por navegador.

5.2.2 Conexões SSL/TLS

A seguir com a análise do *handshake* entre cliente/servidor, o protocolo SSL/TLS foi inspecionado de modo a identificar as informações da camada de segurança do protocolo HTTPS. A Figura 5.6a demonstra o percentual de utilização do protocolo SSL/TLS nos três navegadores por cada sistema operacional. Como se pode observar, o navegador Chrome possui em média 15%, o Firefox 25% e o Opera 14% dos pacotes recolhidos em cada teste. O Firefox se sobressai por utilizar mecanismos próprios de autenticação e certificados digitais o que o torna mais eficiente neste processo. Já os valores aproximados do Chrome e do Opera ocorrem pela semelhança estrutural, conforme citado anteriormente.

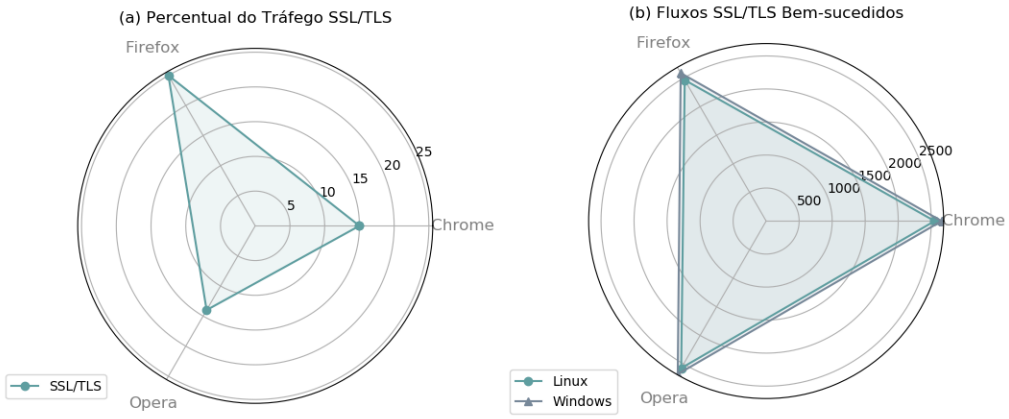

Figura 5.6: Percentual do tráfego SSL/TLS e fluxos bem-sucedidos por cenário.

Outro ponto a destacar é a utilização de extensões próprias da Google para transferência de informações encriptadas que acabam sendo inibidas pelas ferramentas de análise dos dados. Na Figura 5.6b é possível observar que a quantidade de fluxos SSL/TLS ocorridos com *handshake* bem-sucedidos é aproximada em todos os navegadores, mesmo em sistemas operativos distintos.

Mesmo com informações ocultadas pelos navegadores é admissível aceitar que o protocolo trabalha de forma semelhante em qualquer um dos cenários propostos. Na continuação da investigação dos fluxos, foi possível determinar qual a versão do protocolo utilizada tanto pelo cliente quanto pelo servidor. Na Figura 5.7a pode-ser perceber que versão SSL 3.3, composta pelo SSL3 e TLS 1.2 em modo legado (compatível com versões inferiores), é a mais utilizada pela grande maioria dos servidores Web. Já na Figura 5.7b, que representa o lado do cliente, o resultado de que todos os navegadores utilizem a versão SSL3 com TLS 1.0 massivamente surpreendeu. O resultado inesperado demonstra que o protocolo a ser utilizado está obsoleto, além de possuir diversas vulnerabilidades já conhecidas [14].

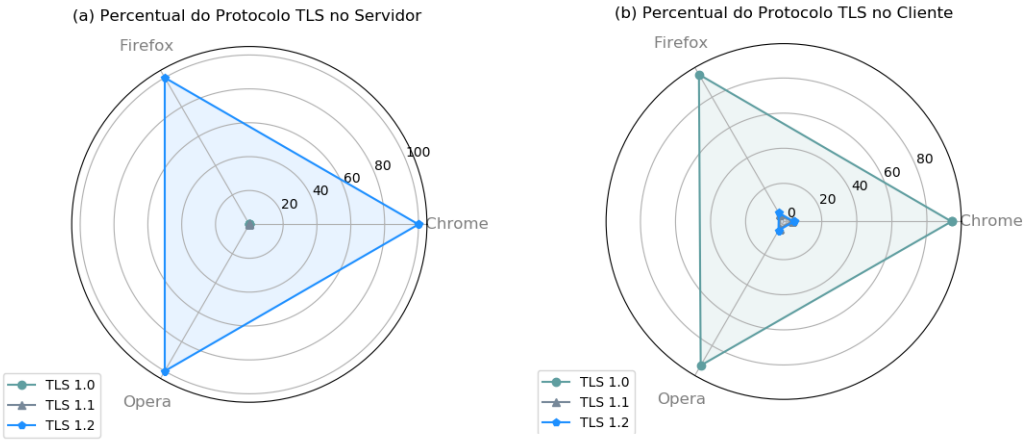

Figura 5.7: Percentual do versão SSL/TLS no lado do Servidor e do Cliente.

Os desenvolvedores dos navegadores estimam que, ainda no ano de 2020, o suporte as versões 1.0/1.1 sejam encerrados [50]. Porém, até o momento de elaboração deste trabalho, os navegadores ainda fazem utilização desta versão como padrão. Para modificar para uma a versão mais atualizada, o utilizador precisa alterar as configurações do navegador de forma manual. Esse tipo de alteração acaba por não ser abrangente, visto que a maioria dos utilizadores não detém esse conhecimento.

Ainda através do fluxos bem-sucedidos, também foi possível verificar mais três tipos de informações: o nome do servidor com o número de cada uma de suas interações, mapeamento das

prováveis atividades do utilizador e os certificados de autenticação. Em relação aos nomes dos servidores é notório, através da tabela 5.2, a visibilidade de informações em um exemplo de interação cliente/servidor pelo navegador Chrome. Pode-se observar a maioria URLs acessadas faz interações com servidores terceiros para recolha de dados com base nos acessos dos utilizadores. Também pode-se verificar um bom número de servidores que direcionam anúncios intrusivos aos utilizadores, ou seja, sem permissão prévia. O programa responsável por essa função é chamado de *adware* e serve para explorar os locais visitados pelo utilizador e redirecioná-lo a publicidade vinculada a sua navegação. As configurações padrões dos navegadores, de acordo com as políticas de privacidade presentes na Sessão 4.2.7, liberam essas informações e o utilizador não possui nenhum tipo de controle a menos que realize modificações nas configurações.

Tabela 5.2: Interações SSL/TLS - Google Chrome.

Servidor	**Tipo**	**Interações por Coleta Diária**
www.facebook.com	Estatístico	67
www.gstatic.com	Anúncios intrusivos	63
a-v2.sndcdn.com	Estatístico	62
www.google.com	Estatístico	52
api-v2.soundcloud.com	Conteúdo	44
www.whiplash.net	Conteúdo	42
www.i.matheranalytics.com	Estatístico	37
nexojornal.s3-sa-east-1.amazonaws.com	Conteúdo	35
ib.adnxs.com	Anúncios intrusivos	31
www.instagram.com	Conteúdo	34
googleads.g.doubleclick.net	Anúncios intrusivos	31
s.yimg.com	Anúncios intrusivos	28
r6—sn-1vo-v2vs.googlevideo.com	Servidor Proxy de conteúdo	27
i.ytimg.com	Anúncios intrusivos	26

No entanto, para o mapeamento das atividades do utilizador, mesmo com os dados do *payload* encriptados, existe a possibilidade de estimar qual acesso foi realizado pelo utilizador apenas com os endereços dos servidores. Por exemplo, na tabela 5.2 o endereço *r6—sn-1vo-v2vs.googlevideo.com* é um servidor Web *Proxy* do serviço de *streaming* de vídeo utilizado por um ISP. Com algumas consultas simples, através páginas Web de geolocalização de endereços IP, foi possível confirmar que o endereço de facto pertence a um ISP de Portugal. Assim, é possível explorar os endereços solicitados pelo navegador e ter conhecimento de toda a navegação realizada.

Também foi possível identificar as datas dos certificados digitais utilizados no processo de autenticação dos fluxos SSL/TLS. A tabela 5.3 oferece um resumo da data de validação dos certificados por endereço acessado, onde os valores são os mesmos para todos os navegadores

utilizados nos testes. Fica evidente que existe uma grande quantidade de certificados que estão fora de validade que podem comprometer a relação de confiança entre cliente/servidor de diversas formas. No entanto, pode observar também que existe uma boa parte dos endereços que estão a utilizar certificados que expirarão nos próximos 50 anos.

Resumidamente, toda informação encontrada no processo de *handshake* antes da encriptação do *payload* demostra exposição a algum tipo de vulnerabilidade. Seja por utilização de um protocolo desatualizado, seja por redirecionamento a endereços intrusivos. Também existe exposição das informações do utilizador que tornam possível a realização de um *footprint*[4] do seu perfil de navegação, como se pode ver em [37] e [38]. Para os certificados digitais inválidos, além de colocar os serviços disponíveis em risco, deixam o utilizador suscetível a fraudes e roubos de identidade [35] [28].

Tabela 5.3: Tempo de validade dos certificados digitais por endereço eletrônico acessado.

URL	**<1990**	**1991-2000**	**2001-2019**	**2020-2050**	**2051-2070**	**>2071**
www.whiplash.net	0%	0%	10%	25%	45%	20%
www.uol.com.br	0%	3%	16%	22%	47%	12%
www.nexojornal.com.br	0%	0%	0%	37%	42%	21%
www.instagram.com.br	0%	10%	14%	26%	39%	11%
www.soundcloud.com	0%	0%	17%	0%	41%	42%
www.youtube.com	0%	0%	12%	36%	27%	25%
www.protonmail.com	9%	12%	0%	79%	0%	0%
www.google.com.br	1%	18%	3%	20%	38%	20%
www.bing.com	0%	0%	0%	0%	45%	55%
www.yahoo.search.com	15%	0%	25%	0%	27%	33%
www.duckduckgo.com	7%	12%	81%	0%	0%	0%
www.gibiru.com	13%	0%	26%	0%	0%	61%
terceiros*	20%	9%	0%	11%	32%	28%

(*) Representa os certificados por URLs de terceiros que atuam no encaminhamento de conteúdo intrusivo.

5.2.3 Cookies

As páginas Web fazem uso de diversos métodos importantes para melhoria da experiência de navegação e entregar um conteúdo interessante para cada visitante. Desta forma, os *cookies* tornam-se uma das técnicas mais conhecidas para essa função conforme visto na Sessão 2.7.1. Nesta etapa foram analisados os seguintes tipos:

[4]Processo de organização de ideias para criar um perfil completo do alvo a ser atacado.

- *cookies de sessão:* armazenados temporariamente no computador do utilizador e são removidos após o fechamento do navegador.

- *cookies persistentes ou de controle:* armazenam informações chaves do utilizador para otimizar sessões futuras entre o cliente/servidor, além de possuir data de validade e ser eliminado automaticamente.

- *cookies de terceiros*: definidos e utilizados por entidades não proprietárias dos *cookies*.

Essas informações sobre *cookies* foram listadas a partir de uma função (*get_cookie*) específica da biblioteca Selenium. Essa função é um pedido em texto pleno sobre as informações enviadas entre cliente/servidor. A seguir, por exemplo, seguem todas as saídas geradas para os três navegadores através do acesso ao portal de notícias Whiplash nos dois cenários utilizados. Neste caso, a única diferença existente entre elas é o identificador do utilizador que é gerado ao inicio de cada sessão.

```
Whiplash portal - Windows/Linux

Chrome
{'_gat_gtag_UA_156193_3': '1', '_gid': 'GA1.2.125023890.1588674206', '_ga':
    'GA1.2.696601717.1588674206', 'paginas_vistas': '1', '__cfduid': '
    d9b106010294c0646c3d4af464ab4b8081588674196'}
Firefox
{'__cfduid': 'd5d8ed5fb200aad1baa6d2bf76a11f6c41588676885', 'paginas_vistas'
    : '1', '_ga': 'GA1.2.1627188287.1588676897',
 '_gid': 'GA1.2.1149617488.1588676897','_gat_gtag_UA_156193_3': '1'}
Opera
{'_gat_gtag_UA_156193_3': '1', '_gid':'GA1.2.490610151.1588679600','_ga':'
    GA1.2.1257842315.1588679600', 'paginas_vistas':'1','__cfduid':'
    d5a0d52900f6f2f19157dc8a1d056356f1588679598'}
```

Nas saídas acima, é possível identificar alguns tipos de *cookies*:

- *_gat_gtag:* associado ao Google Analytics[5] para fixar a taxa de solicitações e limitar a coleta de dados em sites de alto tráfego. E expira após 10 minutos;

[5]Serviço da Google para monitoramento de tráfego que pode ser agregado a qualquer endereço eletrônico e ferramenta fundamental para o marketing digital.

- *_ga:* também e associado ao Google Analytics para atualização dos serviços de análise utilizados pelo Google. Faz distinção de utilizadores únicos com a geração de uma identificação de clientes. E é incluído em requisições cliente/servidor para calcular dados dos visitantes, sessões e gerar relatórios de análises em uma página Web. Por padrão pode expirar em 2 anos, mas pode ser modificado pelo proprietário do serviço;
- *_gid:* associado ao Google Analytics, parece armazenar e atualizar um valor exclusivo de identificação para cada página visitada em um servidor Web. Este *cookie* expira em 24 horas;
- *__cfuid:* associado a Cloudfare[6] para maximizar os recursos da rede, gerenciar o tráfego e proteger os sites contra tráfego mal-intencionado. O seu tempo de vida é no máximo de 30 minutos.

A partir dos dados recolhidos no processo de simulação, foi possível gerar a tabela 5.4 que sumariza os tipos de *cookies* e suas saídas encontradas em todas URLs acessados. É possível notar que todos os endereços armazenam poucos *cookies* de sessão, que costumam ser removidos após o fechamento do navegador. Os persistentes são utilizados pela grande maioria dos endereços listados, o que deixa claro que há preferência por conservar os dados dos utilizadores. Também é possível observar que a maior parte dos endereços trabalha com *cookies* de terceiros, ou seja, fazem compartilhamento de informações para compor diversas bases de dados e anúncios intrusivos. O exemplo dos *cookies* retirados do portal Whiplash demonstra que as informações, de facto, são compartilhadas e disponibilizadas em forma de estatísticas. Outro destaque é para o portal UOL, cujo URL é o que mais possui *cookies* persistentes. Como é um portal com milhares de acessos mensais, torna-se um endereço eletrônico atrativo para mapear diversas informações sobre os utilizadores em larga escala.

O tempo de expiração dos certificados serve para ter uma noção do período que os dados dos utilizadores são armazenados por uma base de dados. Essa constatação configura um alerta sobre a possibilidade da existência de abusos praticados pelas políticas de segurança adotadas por navegadores, motores de busca e páginas Web. Essa questão ocorre devido às informações do período da retenção de dados não serem claras em nenhum dos casos. Isto é, o utilizador não tem nenhuma noção para quem ou por quanto tempo seus dados ficarão realmente disponíveis na Internet. Desta forma, a adoção de melhores práticas que visem a redução desse tempo devem ser adotadas com mais transparência. Outra questão é que a falta de controle sobre os arquivos *cookies* pode ocasionar roubo de informações acaso não apresentem códigos de autenticação corretos, pois há diversos estudos que comprovam essas falhas [16]. Para esse estudo, a maioria das sessões estavam autenticadas e aparentemente não apresentaram a existência de vulnerabilidades.

[6]Empresa que oferece um conjunto de serviços de proteção e otimização de páginas Web.

Tabela 5.4: Tipo de cookies encontrados por endereço eletrônico.

Endereço Eletrônico	**Cookies de Terceiros**	**Cookies Persistentes**	**Cookies de Sessão**
www.whiplash.net	4	8	2
www.uol.com.br	6	24	9
www.nexojornal.com.br	1	5	2
www.instagram.com	1	6	0
www.soundcloud.com	0	1	0
www.youtube.com	1	3	2
www.protonmail.com	0	2	0
www.google.com	1	5	0
www.bing.com	2	7	2
www.yahoo.search.com	1	4	2
www.duckduckgo.com	0	0	0
www.gibiru.com	0	0	0
Média Geral	1.42	5.5	1.6

Em seguida buscou-se verificar o tempo de expiração dos *cookies* encontrados. Os dados foram concentrados na tabela 5.5, onde é possível verificar que o tempo expiração mais utilizado para armazenamento é o período de 2 anos. Porém, foram encontrados outros endereços que concentra-se em armazenar informações por muito mais tempo, com variações que podem ser de 5 dias até 20 anos. São poucos os endereços que fazem uso de dados que expiram em menos tempo com destaque para os *cookies* que são descartados em até 10 minutos ou quando o navegador é simplesmente fechado.

Tabela 5.5: Percentual de cookies por tempo de validade.

Tempo expiração	**Percentual**
10 minutos ou quando o navegador for finalizado	17,5%
Até 24 horas	6,0%
Até 5 dias	4,5%
Até 30 dias	11,0%
Até 100 dias	8,0%
Até 300 dias	2,0%
Até 1 ano	7,8%
Até 2 anos	36,0%
Até 10 anos	6,2%
Até 20 anos	1,0%

5.2.4 HTTP e HTTPS

Em termos de adoção da versão do protocolo HTTP, identificou-se o uso expressivo da versão 1.1, como pode ser visto na Figura 5.8. Mesmo com o HTTP/2 em crescente adoção e com os navegadores configurados por padrão para uso automático desse protocolo, não houve registros de versões diferentes da versão 1.1 em nenhum dos dois cenários de testes. No entanto, existem estudos que comprovam que a adesão é uma realidade por diversos servidores Web [31] [32].

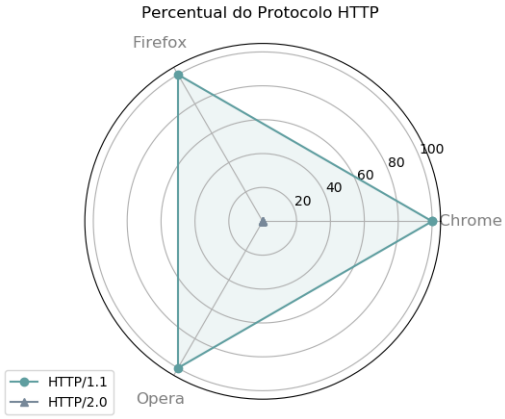

Figura 5.8: Percentual do protocolo HTTP/1.1.

Outra análise realizada foi a utilização do protocolo HTTPS em relação ao HTTP. A Figura 5.9a sumariza os resultados do sistema operacional Linux. O Firefox detém mais de 93% do seu tráfego encriptado. Já o Chorme tem um comportamento diferente em relação aos outros navegadores, pois apenas 59% é considerado como HTTPS e 29,7% HTTP. O alto índice do uso do HTTP, neste caso, ocorre devido ao navegador utilizar o protocolo QUIC nas conexões TCP realizadas por este mesmo protocolo. Conforme já citado anteriormente, esse protocolo é proprietário do Google e também utiliza SSL/TLS para encriptar a troca de dados. O Opera também apresenta essa mesma particularidade com o QUIC devido a utilizar o mesmo motor de compilação que o Chrome. Desta forma, apresenta 49% para HTTPS e 26% para HTTP. A Figura 5.10 expõe um exemplo de requisição bem sucedida entre cliente/servidor com evidência na presença do protocolo na porta 443 para os navegadores Chrome e Opera.

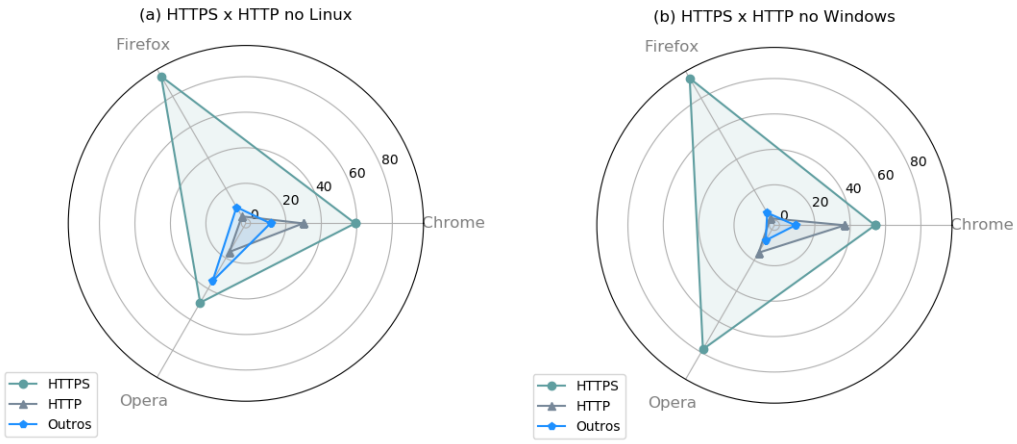

Figura 5.9: Percentual de Utilização HTTP e HTTPS no Linux e Windows.

Na Figura 5.9b são demonstrados os resultados para o sistema operacional Windows e como pode-se observar eles são semelhantes com Opera a apresentar maior uso de HTTPS em Windows. Portanto, se constata que o comportamento dos navegadores é similar em todos os cenários de teste e com percentuais altos de encriptação de informações. O sistema operacional, nesse caso, não é um fator determinante para melhor utilização de um protocolo de navegação seguro para o utilizador final.

```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Disposition: attachment
Content-Length: 9192632
Content-Security-Policy: default-src 'none'
Content-Type: application/octet-stream
Etag: "588fea"
Server: downloads
Vary: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 0
Date: Mon, 04 May 2020 19:10:22 GMT
Alt-Svc: h3-Q050=":443"; ma=2592000,h3-Q049=":443"; ma=2592000,h3-Q048=":443";
ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
Last-Modified: Thu, 02 Apr 2020 18:14:12 GMT
Connection: keep-alive
```

Figura 5.10: Requisição de Resposta HTTP.

Mesmo com adoção em larga escala do HTTPS, ainda existem muitos acessos que não utilizam nenhum tipo de segurança. Para este trabalho, todos URLs selecionadas utilizam protocolos de comunicação segura entre cliente/servidor. Entretanto, isso não significa que o protocolo HTTPS não apresente falhas, como demonstrado em trabalhos como [35] e [14]. Ainda, mesmo com a encriptação dos dados, é possível captar diversas informações dos utilizadores das mais diversas

formas, nomeadamente o tamanho de pacotes [25], *machine learning* [29], reconstrução de conexões TCP [30] ao *handshake* SSL/TLS [33].

5.2.5 Comportamento dos Navegadores

Os navegadores selecionados dispõem de uma série de recursos de configurações que vão da melhoria de recursos a restrição do compartilhamento de informações. Muitos utilizadores não possuem um conhecimento avançado de informática para customizar um navegador para suas necessidades de navegação, pois na maioria dos casos almejam apenas que o processo de navegação seja eficiente. Desta maneira, muitos navegadores apresentam uma configuração padrão ao serem instalados nos dispositivos. Essas configurações iniciais são baseadas na politicas de privacidade aceites pelo utilizador na instalação do navegador, visto que esse processo só é concluído com a aceitação de todos os termos propostos.

A tabela 5.6 resume algumas das configurações presentes nos navegadores Chrome, Firefox e Opera. É válido ressaltar que as configurações não sofrem alterações pela diferença dos sistemas operativos dos cenários de testes. A tabela evidencia que a maioria dos parâmetros encontra-se ativa e está de acordo com as políticas de privacidade citadas na Sessão 4.1.6.

Tabela 5.6: Configurações iniciais dos navegadores.

Configurações	**Chrome**	**Firefox**	**Opera**
Integração com conta de serviços próprios	Ativado	Ativado	Ativado
Atualizações automáticas	Ativado	Ativado	Ativado
Bloqueadores de conteúdo malicioso	Ativado	Ativado	Ativado
Registro e manipulação de cookies	Ativado	Ativado	Ativado
Deletar registros de cookies ao fechar o navegador	Desativado	Desativado	Desativado
Salvar senhas/Logins	Ativado	Ativado	Ativado
Salvar histórico de pagamentos	Ativado	Ativado	Ativado
Salvar histórico de pagamentos com criptomoedas	Ativado	Ativado	Ativado
Salvar histórico de navegação	Ativado	Ativado	Ativado
Localização do utilizador	Ativado	Ativado	Ativado
Coleta de dados técnicos	Ativado	Ativado	Ativado
Coleta de dados do utilizador	Ativado	Ativado	Ativado

Fica evidente a quantidade de informação que o utilizador disponibiliza para os navegadores, pois toda sua atividade está a ser mapeada através de sua localização, históricos de pagamentos, histórico de acessos, senhas e diversos outros dados sensíveis. Outro destaque vai para o processo de registro, manipulação e não remoção dos *cookies* utilizados, que também pode vir a apresentar um alerta sobre o uso abusivo de políticas de segurança e carece de mais aprofundamento. Em

questões de segurança de acesso pode-se verificar que os navegadores são compostos de critérios para bloqueio a endereços maliciosos.

Essas configurações são apenas modificações simples que podem vir a ser alteradas por qualquer utilizador. Entretanto, todos os navegadores possuem módulos mais avançados que podem restringir a maneira que a recolha de dados é realizada e até inserção de novos recursos. Esses parâmetros não foram modificados e nem testados, mas o registro é importante uma vez que possibilita a utilização de funções experimentais como o TLS na versão 1.3 e o protocolo QUIC como protocolo padrão.

Também foi realizado o mapeamento do volume de dados gerados por cada navegador durante a simulação. O Chrome demonstrou uma média maior do volume de dados com mais de 211 MB. Já o Firefox e o Opera tiveram 162 e 157 MB, respectivamente. Como pode-se notar na Figura 5.11, é uma diferença de mais de 50 MB. Os dois sistemas operativos utilizados não apresentam quaisquer diferenças nos cenários testados.

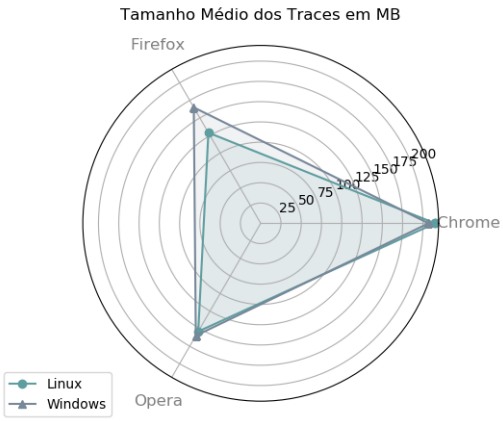

Figura 5.11: Tamanho médio do tráfego por Navegador.

Através do fluxo de *handshake*, buscou-se estimar o motivo pelo qual o Chrome possui quantidades maiores de tráfego. Como os navegadores fazem comunicação constante com suas bases de dados, o tamanho do corpo de resposta do HTTPS foi utilizado como parâmetro de identificação. Na tabela 5.7 é possível ver quais os servidores Web que mais obtiveram respostas dos navegadores e o tamanho do cabeçalho de resposta. O Firefox e o Opera atuam com dois servidores e com cabeçalhos na faixa dos 600 bytes por servidor. Já o Chrome que possui quatro servidores e com cabeçalhos na faixa dos 563 bytes, o que induz a possuir um número maior de interações e de volume de dados.

Tabela 5.7: Cabeçalhos de resposta HTTPS por navegador.

Navegador	**Servidores**	**Resposta do cabeçalho em Bytes**	**Média de Interações por Servidor**
Firefox	firefox.settings.services.mozilla.com	590	130
	content-signature-2.cdn.mozilla.net	590	60
Chrome	www.google.com	563	71
	www.google-analytics.com	563	64
	update.googleapis.com	563	38
	www.gstatic.com	563	42
Opera	exchange.opera.com	563	107
	af.opera.com	563	102

Em relação a identificação de dados do utilizador, como sistema operacional e versão do navegador, é possível visualizar em ambos os cenários. A Figura 5.12 mostra uma captura das informações visíveis pelo navegador em cada sistema operacional.

(a) Registro dos navegadores no Linux

```
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 Chrome/81.0.4044.129
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:75.0) Gecko/20100101 Firefox/75.0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 Chrome/81.0.4044.129 OPR/68.0.3618.63
```

(b) Registro dos navegadores no Windows

```
User-Agent: Microsoft BITS/7.8 AppleWebKit/537.36 Chrome/81.0.4044.129
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:75.0) Gecko/20100101 Firefox/75.0
User-Agent: Microsoft-CryptoAPI/10.0 AppleWebKit/537.36 Chrome/81.0.4044.129 OPR/68.0.3618.63
```

Figura 5.12: Informações dos navegadores no Linux e no Windows.

Mesmo com a informação que os seus dados são recolhidos, o utilizador não tem controle do que é feito e nem para onde os seus dados são realmente enviados. A exposição de suas informações é um facto e se mostra um perigo real a diversos tipos de ataques e vulnerabilidades [8]. O Regulamento Geral de Proteção de Dados, ou simplesmente RGPD[7], foi desenvolvido para proteção do utilizador em relação ao tratamento dos seus dados pessoais em grande escala pelas mais diversas empresas e serviços.

[7]Mais informações em https://rgpd-portugal.pt/

5.2.6 Motores de Busca

Os motores de busca são ferramentas importantes que auxiliam os utilizadores a realizarem pesquisas sobre os mais diversos assuntos. Diante disso, o comportamento dos motores de busca foi analisado para se verificar se o redirecionamento para conteúdos seguros se dá de forma eficiente a partir dos navegadores elegidos. Conforme as políticas de privacidade apresentadas na Sessão 4.1.7, todos os motores de busca se propõem a redirecionar os utilizadores para apenas conteúdos que utilizam o protocolo HTTPS.

Para esse teste foram escolhidas algumas palavras que estão em evidência no momento em que este trabalho foi escrito. As palavras selecionadas foram: covid-19, educação, empregos, tecnologia, teleaula e teletrabalho. Para a escolha de um termo antes do processo de busca, utilizou-se um artifício da biblioteca Selenium que permite uma seleção aleatória e o repasse da palavra elegida para o campo de pesquisa.

Na Figura 5.13 pode ser observado que os motores de busca, em ambos cenários, têm valores aproximados para a média de encaminhamento de endereços que não utilizam comunicação HTTPS. Isto é, conteúdo dito como não seguro e apresentado em HTTP. Para os motores de busca mais acessados pelos utilizadores, tem destaque o Yahoo que encaminha menos de 3% das suas pesquisas para endereços não seguros em todos os navegadores. Em seguida se destacam o Bing com 5% e o Google com 6%. Já os motores de busca alternativos apresentam diferenças substanciais entre os navegadores. O DuckDuckGo tem uma média 13% das pesquisas não seguras para Chrome e Firefox, já o Opera ultrapassou os 20% que se repete nos dois cenários de testes. O Gibiru possui o mesmo comportamento do anterior com 12% para o Chrome, 14% no Firefox e mais de 15% no Opera.

Esses valores estão relacionados com a política de *cookies* utilizada por cada motor de busca e evidenciada na tabela 5.4. Aparentemente, se um motor de busca consegue armazenar informações do navegador e do utilizador, ele conseguirá fazer redirecionamentos mais eficientes. Também pode-se levar em consideração que os motores de busca mais conhecidos têm políticas de busca patrocinada, onde qualquer endereço eletrônico pode pagar taxas para aparecer como pesquisa relevante.

Ao analisar a política de privacidade do Yahoo, por exemplo, pode-se identificar que ele manipula e adiciona informações no computador do utilizador. Consequentemente, possui o índice mais baixo de redirecionamento para conteúdos inseguros.

Em relação aos motores de busca alternativos, que não fazem nenhum tipo de controle de *cookies*, apresentam maiores índices de pesquisas para endereços com conteúdo sem segurança. Como também não possuem busca patrocinada, retornam mais endereços inseguros e muitas

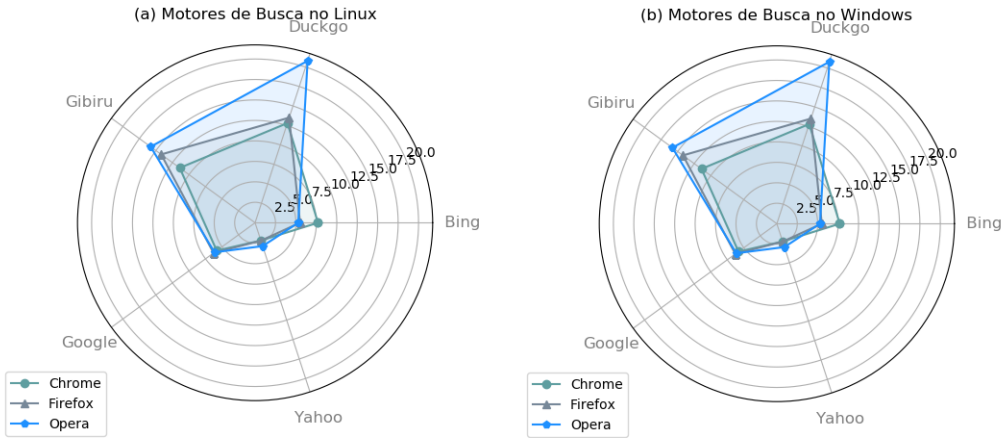

Figura 5.13: Percentual de redirecionamento para URLs seguras dos motores de busca no Linux e no Windows.

vezes com conteúdo mais antigo.

Outra característica observada nos motores de busca é que quanto mais comum o termo que for pesquisado, mais fácil é de retornar endereços que não sejam seguros. Na Figura 5.14 é possível verificar que incidência de pesquisa para o termo Covid-19, que é um termo mais técnico para buscas, tem 4% de encaminhamento para endereços não seguros. Já para termos como educação e empregos chegam até 18% e 17% respectivamente, para conteúdo sem segurança.

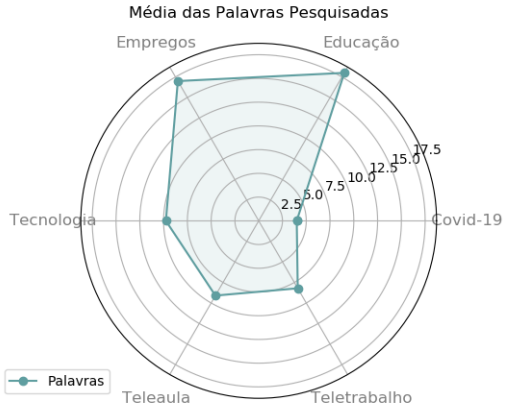

Figura 5.14: Médias das palavras pesquisadas nos dois cenários.

Portanto, é possível identificar que os motores de busca apresentam prováveis vulnerabilidades em relação a resultados de pesquisas sem a segurança do protocolo de comunicação HTTPS.

Também reforça a evidência de mais um ponto de exposição de dados dos utilizadores.

5.3 Síntese dos Resultados

Esta Sessão reúne uma breve abordagem dos dados encontrados na análise de todo o tráfego analisado.

Na etapa de Conexões TCP foi identificado que todos os navegadores fazem utilização de recursos de otimização de conexões, uso massivo do IPv4, visibilidade de portas e uso automático do SSDP para localização de dispositivos. O destaque dessa etapa vai para o Firefox, que se sobressai em relação ao Chrome e ao Opera, ao suprimir a maior quantidade informações da camada de rede de modo a diminuir vulnerabilidades, além de não fazer uso do protocolo SSDP.

Para a etapa de Conexões SSL/TLS, verificou-se, através do processo de *hansdhake* entre cliente/servidor, o percentual de uso de protocolo por navegador, existência de versões obsoletas do protocolo TLS, possibilidade de identificar informações dos servidores Web acessados de modo a permitir o mapeamento de acesso do utilizador, exposição de dados dos utilizadores e certificados digitais desatualizados. Nessa etapa todos os navegadores apresentam possíveis vulnerabilidades e exposição de dados para todos os itens citados.

Em relação aos *Cookies*, que é a terceira etapa, observou-se a quantidade de informações que são coletadas e compartilhadas pelos servidores Web, o tempo em que esses dados podem ficar armazenados que varia de 10 minutos a 20 anos, falta de transparência sobre a manipulação desses dados por parte dos navegadores e servidores Web e exposição de informações para diversas bases de armazenamento de dados. Nesta etapa também todos os navegadores têm o mesmo comportamento em relação exposição de dados.

A etapa HTTP e HTTPS demonstra o uso massivo do HTTP1.1 e também que todos os navegadores já utilizam o HTTPS como principal protocolo de comunicação segura. Também é perceptível o uso do protocolo QUIC pelos navegadores Chrome e Opera para melhorar o transporte de dados encriptados. Porém, esse protocolo é utilizado com algumas extensões proprietárias e encriptadas do Google e não possibilita identificar em detalhe o seu uso sobre HTTP. O destaque fica com Firefox, que consegue utilizar mais de 90% do seu tráfego com HTTPS. Mesmo com números elevados, a exposição de informações de conexões cliente/servidor são visíveis em todos os navegadores.

Na etapa de Comportamento dos Navegadores se evidencia quais configurações iniciais são impostas pelos navegadores e como elas refletem na exposição de dados do utilizador. Foi constatado que todos os navegadores fazem a exposição e compartilhamento das informações dos utilizadores,

com destaque para o Chrome por ser o navegador que mais recolhe dados dos utilizadores.

A última etapa, Motores de Busca, demonstra o comportamento na pesquisa de um termo qualquer pelo utilizador. Através de um grupo de palavras observou-se que termos comuns são mais propícios a serem redirecionados para endereços de conteúdo não seguro. Também é observado que a coleta de *cookies* está diretamente ligada a resultados seguros, ou seja, quanto mais intrusivo for o *cookie* mais resultados seguros são selecionados. Também não é possível realizar um destaque de um Motor de Busca, visto que todos possuem possíveis vulnerabilidades e exposição dos dados.

A tabela 5.8 sumariza todos as informações encontradas relacionadas aos navegadores e a tabela 5.9 aos motores de busca.

Tabela 5.8: Resumo dos dados analisados dos navegadores.

Etapa de Investigação	Característica	Classificação	Chrome	Firefox	Opera
Conexões TCP					
Otimização de recursos	-	E	Sim	Sim	Sim
Protocolo SSDP	V	E	Sim	Não	Sim
Portas visíeis	V/ED	E	Sim	Não	Sim
Uso IPv4	V	E	Sim	Sim	Sim
SSL/TLS					
Informações utilizadores	ED	E	Sim	Sim	Sim
Informações servidores Web	ED	E	Sim	Sim	Sim
Compartilhamento de Informações	ED	E	Sim	Sim	Sim
Certificados digitais vencidos	V	E	Sim	Sim	Sim
Protocolo TLS obsoleto	V	E	Sim	Sim	Sim
Cookies					
Persistentes	ED	E	Sim	Sim	Sim
Armazenamento de informações	ED	E	Sim	Sim	Sim
Compartilhamento de informações	ED	E	Sim	Sim	Sim
Validade entre 10min a 20 anos	ED	E	Sim	Sim	Sim
HTTP e HTTPS					
Encriptação dos dados	TE	E	Sim	Sim	Sim
Exposição de dados da conexão	V/ED/TE	E	Sim	Sim	Sim
Navegadores					
Exposição de dados sensíveis	V/ED	E	Sim	Sim	Sim
Coleta e armazenamento de dados	ED	E	Sim	Sim	Sim
Compartilhamento dos dados	ED	E	Sim	Sim	Sim
Bloqueio de endereços maliciosos	-	E	Sim	Sim	Sim

Tabela 5.9: Resumo dos dados analisados dos motores de busca.

Etapa de Investigação	Caract.	Class.	Bing	Google	Yahoo	DuckGo	Gibiru
Motores de Busca							
Exposição de dados sensíveis	ED	E	Sim	Sim	Sim	Sim	Sim
Coleta e armazenamento de dados	ED	E	Sim	Sim	Sim	Não	Não
Compartilhamento dos dados	ED	E	Sim	Sim	Sim	Não	Não
Redirecionamento eficiente	TE	E	Sim	Sim	Sim	Não	Não
Resultados sem datas	V	E	Não	Não	Não	Sim	Sim

(*) Para as características de verificação de cada etapa, serão adotados os seguintes critérios: ED - Exposição de dados, V - Vulnerabilidades e TE - Tráfego Encriptado.
(**) Para os níveis de classificação são adotados os critérios: E - Existente e I - Inexistente.

De uma forma geral conclui-se:

- o comportamento de navegação é o mesmo ao comparar os sistemas operativos Linux e Windows;
- todos os navegadores possuem algum tipo de vulnerabilidade;
- todos os navegadores expõem dados sensíveis dos utilizadores;
- todos URLs e motores de busca também expõem os dados dos utilizadores;
- todos os navegadores, URLs e motores de busca armazenam e compartilham as informações recolhidas dos utilizadores com endereços terceiros;
- possibilidade de abusos nas políticas de privacidade aceites pelos utilizadores;
- todos os navegadores fazem uso de protocolos de comunicação segura.

5.4 Sumário

A primeira parte do capítulo apresentou a taxonomia para realização dos testes, de forma que os resultados da metodologia exposta no Capítulo 4 fossem evidenciados. A segunda parte foi responsável por mostrar uma análise de todo o tráfego gerado pela simulação de uma navegação na Internet. A análise buscou apresentar as diferenças de acesso entre dois sistemas operativos,

com três tipos de navegadores e vários motores de busca para evidenciar as mudanças do processo de coleta e exposição dos dados do utilizador na experiência de navegação na Internet com a implantação da comunicação segura entre cliente e servidor.

Capítulo 6

Conclusões

Neste capítulo apresenta-se um resumo do trabalho desenvolvido e discutido ao longo desta dissertação. Apresentam-se também temas possíveis para serem desenvolvidos em trabalhos futuros.

6.1 Resumo do Trabalho Desenvolvido

As ferramentas específicas que procuram caracterizar o protocolo HTTPS, objeto de estudo dessa dissertação, foram aplicadas através da medição passiva de tráfego na qual os dados são armazenados para depois serem tratados. A utilização de diversas ferramentas foi necessária para ter perspectivas diferentes da mesma informação, de maneira a possibilitar a coleta de dados relevantes, já que o uso de apenas uma única ferramenta pode apresentar limitações quanto aos resultados. A junção do Tstat, com as ferramentas Libtrace, Libprotoident, SSLAnalyzer e Fiddler, que utilizam DPI e LPI por exemplo, foi essencial para o mapeamento de dados através de técnicas de extração de assinaturas do protocolo SSL/TLS durante o estabelecimento da conexão segura entre cliente/servidor.

O primeiro objetivo deste trabalho procurou analisar quais os impactos significativos promovidos pelo protocolo HTTPS na conexão do utilizador através de uma simulação do processo de navegação. Pelas observações realizadas, pode-se concluir que o utilizador não percebe as alterações na sua forma de acesso. Isso ocorre porque a maioria das questões que envolvem o protocolo HTTPS não estão facilmente acessíveis e a falta de conhecimento técnico por parte do utilizador comum para buscá-las contribui para que essa informação não seja percebida. Algumas aplicações específicas visam prestar um auxílio para que essa informação chegue ao utilizador mas, por vezes,

este apenas se preocupa em ter uma conexão eficiente.

Constata-se que a navegação se tornou realmente segura e existe uma preocupação em proteger a conexão com servidores Web, as informações dos utilizadores e o conteúdo acessado de atores externos. Mas por outro lado, essas informações acabam por ser coletadas, armazenadas e compartilhadas por navegadores, endereços eletrônicos e motores de busca sem nenhum tipo de controle por parte do utilizador. A conexão passou a ser segura, mas a exposição das informações dos utilizadores aumentou significativamente. Também, as políticas de privacidade adotadas aparentam ser abusivas e muitas vezes o utilizador aceita os termos sem ter noção do que está a autorizar.

Dito isso, é válido ressaltar que algumas informações encontradas no processo de comunicação cliente/servidor, antes da encriptação dos dados, demostram algum tipo de vulnerabilidade e exposição de dados sensíveis do utilizador. Seja por utilização de um protocolo desatualizado, seja por redirecionamento a endereços intrusivos ou até mesmo por exposição de portas de comunicação. Outro ponto é a presença de certificados digitais inválidos, que além de colocar os serviços disponíveis em risco, deixam o utilizador suscetível a possíveis fraudes e roubos de identidade.

O segundo objetivo visava verificar se os sites de busca realizam redirecionamento dos utilizadores de forma efetiva para endereços mais seguros. Conclui-se que a utilização de *cookies* está ligado diretamente a esse processo, visto que quanto maior a manipulação por parte de navegadores e motores de busca, maior é a quantidade de endereços seguros ao qual o utilizador é redirecionado. Também é possível identificar que os motores de busca apresentam prováveis vulnerabilidades em relação a resultados de pesquisas sem a segurança do protocolo de comunicação HTTPS e reforça a evidência de mais um ponto de exposição de dados dos utilizadores.

O terceiro objetivo consistia identificar e comparar comportamentos no acesso realizado pelo utilizador a partir de um computador pessoal com sistemas operacionais diferentes. No entanto, conclui-se que para sistemas operativos Linux e Windows não existem diferenças ou impactos no processo de navegação.

Para além do cumprimento dos objetivo estabelecidos, este projeto também permitiu alertar para a existência de uma grande exposição dos dados do utilizador. Além da exposição, é notado o compartilhamento com diversas bases de armazenamento de dados e como mencionado anteriormente, o utilizador não tem conhecimento do destino ao qual seus dados são enviados, para quem e nem por quanto tempo eles estarão disponíveis. Surge então a questão: De que adianta proteger a conexão, se tantas outras informações são disponibilizadas sem controle? Diante disso, a RGPD torna-se uma forma essencial para o utilizador combater essa grande exposição de suas informações.

6.2 Trabalhos Futuros

Com relação a trabalhos futuros sugere-se a ampliação do número de navegadores de forma a identificar se o comportamento se assemelha aos que foram analisados (*Chrome, Firefox e Opera*).

Propõe-se também a investigação em sistemas operacionais de dispositivos móveis, como o Android e o IOS, e a verificação se existem diferenças significativas dos resultados em comparação aos sistemas operacionais de computadores.

Outra proposta é a utilização de ferramentas para identificação de tráfego cifrado no payload e expandir a pesquisa para além da visualização das informações no processo de comunicação das máquinas cliente/servidor.

Pretende-se também realizar a diversificação de cenários de navegação, com um processo aleatório de acesso a URL's, para cobrir os diferentes tipos perfis de utilizadores e um período de medição mais extenso para uma base mais ampla de informações.

Por fim, sugere-se a utilização de ferramentas de *machine learning* na tentativa de automatizar a investigação de mais informações que estão a ser expostas e a criação de um dicionário padrão de exposição de dados para esse tipo de pesquisa.

Apêndices

Apêndice A

TSTAT

A.1 Core TCP Set

C2S	S2C	Short description	Unit	Long description
1	15	Client/Server IP addr	-	IP addresses of the client/server
2	16	Client/Server TCP port	-	TCP port addresses for the client/server
3	17	packets	-	total number of packets observed form the client/server
4	18	RST sent	0/1	0 = no RST segment has been sent by the client/server
5	19	ACK sent	-	number of segments with the ACK field set to 1
6	20	PURE ACK sent	-	number of segments with ACK field set to 1 and no data
7	21	unique bytes	bytes	number of bytes sent in the payload
8	22	data pkts	-	number of segments with payload
9	23	data bytes	bytes	number of bytes transmitted in the payload, including retransmissions
10	24	rexmit pkts	-	number of retransmitted segments
11	25	rexmit bytes	bytes	number of retransmitted bytes
12	26	out seq pkts	-	number of segments observed out of sequence
13	27	SYN count	-	number of SYN segments observed (including rtx)
14	28	FIN count	-	number of FIN segments observed (including rtx)
29		First time abs	ms	Flow first packet absolute time (epoch)
30		Last time abs	ms	Flow last segment absolute time (epoch)
31		Completion time	ms	Flow duration since first packet to last packet
32		C first payload	ms	Client first segment with payload since the first flow segment
33		S first payload	ms	Server first segment with payload since the first flow segment
34		C last payload	ms	Client last segment with payload since the first flow segment
35		S last payload	ms	Server last segment with payload since the first flow segment
36		C first ack	ms	Client first ACK segment (without SYN) since the first flow segment
37		S first ack	ms	Server first ACK segment (without SYN) since the first flow segment
38		C Internal	0/1	1 = client has internal IP, 0 = client has external IP
39		S Internal	0/1	1 = server has internal IP, 0 = server has external IP
40		C anonymized	0/1	1 = client IP is CryptoPAn anonymized
41		S anonymized	0/1	1 = server IP is CryptoPAn anonymized
42		Connection type	-	Bitmap stating the connection type as identified by TCPL7 inspection engine (see protocol.h)
43		P2P type	-	Type of P2P protocol, as identified by the IPP2P engine (see ipp2p_tstat.h)
44		HTTP type	-	For HTTP flows, the identified Web2.0 content (see the http_content enum in struct.h)

Figura A.1: Saídas Core TCP Set.

A.2 TCP Layer 7 Set

C2S	S2C	Short description	Unit	Long description
K		HTTP Request count	-	Number of HTTP Requests (GET/POST/HEAD) seen in the C2S direction (for HTTP connections)
K+1		HTTP Response count	-	Number of HTTP Responses (HTTP) seen in the S2C direction (for HTTP connections)
K+2		First HTTP Response	-	First HTTP Response code seen in the server->client communication (for HTTP connections)
K+3		PSH-separated C2S	-	number of push separated messages C2S
K+4		PSH-separated S2C	-	number of push separated messages S2C
K+5		TLS Client Hello SNI	-	For TLS flows, the server name indicated by the client in the Hello message extensions
K+6		TLS Server Hello SCN	-	For TLS flows, the subject CN name indicated by the server in its certificate
K+7		TLS Client NPN/ALPN	-	For TLS flows, a bitmap representing the usage of NPN/ALPN for HTTP2/SPDY negotiation
K+8		TLS Server NPN/ALPN	-	For TLS flows, a bitmap representing the usage of NPN/ALPN for HTTP2/SPDY negotiation
K+9		TLS Client ID reuse	-	For TLS flows, indicates that the Client Hello carries an old Session ID
K+10		TLS Client Last Handshake	ms	For TLS flows, time of Client last packet seen before first Application Data (relative)
K+11		TLS Server Last Handshake	ms	For TLS flows, time of Server last packet seen before first Application Data (relative)
K+12		TLS Client App Data Time	ms	For TLS flows, time between the Client first Application Data message and the first flow segment
K+13		TLS Server App Data Time	ms	For TLS flows, time between the Server first Application Data message and the first flow segment
K+14		TLS Client App Data Bytes	bytes	For TLS flows, relative sequence number for the Client first Application Data message
K+15		TLS Server App Data Bytes	bytes	For TLS flows, relative sequence number for the Client first Application Data message
K+16		FQDN	-	Fully Qualified Domain Name recovered using DNHunter
K+17		IP of DNS resolver	-	IP address of the contacted DNS resolver
K+18		DNS request time	ms	unixtime (in ms) of the DNS request
K+19		DNS response time	ms	unixtime (in ms) of the DNS response

Figura A.2: Saídas TCP Layer 7 Set.

A.3 Coluna 42 - Protocolos Identificados

Connection type - col.42 (see protocol.h)	
Bitmask Value	**Protocol**
0	Unknown protocol
1	HTTP protocol
2	RTSP protocol
4	RTP protocol
8	ICY protocol
16	RTCP protocol
32	MSN protocol
64	YMSG protocol
128	XMPP protocol
256	P2P protocol
512	SKYPE protocol
1024	SMTP protocol
2048	POP3 protocol
4096	IMAP4 protocol
8192	TLS/TLS protocol
16384	ED2K protocol (obfuscated)
32768	SSH 2.0/1.99 protocol
65536	RTMP protocol
131072	Bittorrent MSE/PE protocol

Figura A.3: Lista de protocolos de aplicação.

A.4 Log HTTP Complete

C2S	S2C	Short description	Unit	Long description
1	1	Client IP addr	-	IP addresses of the client (sending the request/receiving the response)
2	2	Client TCP port	-	TCP port addresses for the client
3	3	Server IP addr	-	IP addresses of the server (receiving the request/sending the response)
4	4	Server TCP port	-	TCP port addresses for the server
5	5	Segment time abs	s	Absolute time [s] (epoch) of the request/response
6		Request method	-	Request method (GET/POST/HEAD) [*]
7		Hostname	-	Value of the "Host:" HTTP request field
8		FQDN	-	DN-Hunter cached DNS name [^]
9		URL Path	-	URL request path
10		Referer	-	Value of the "Referer:" HTTP request field
11		User agent	-	Value of the "User-Agent:" HTTP request field
12		Cookie	-	Value of the "Cookie:" HTTP request field
13		Do Not Track	-	Value of the "DNT:" HTTP request field
	6	Response string	-	Response identifier (always "HTTP") [*]
	7	Response code	-	HTTP response code (2xx/3xx/4xx/5xx)
	8	Content len	bytes	Value of the "Content-Length:" HTTP response field
	9	Content type	-	Value of the "Content-Type:" HTTP response field
	10	Server	-	Value of the "Server:" HTTP response field
	11	Range	-	Value of the "Content-Range:" HTTP response field for partial content (Code 206)
	12	Location	-	Value of the "Location:" HTTP response field for redirected content (Code 302)
	13	Set Cookie	-	Value of the "Set-Cookie:" HTTP response field

Figura A.4: Saídas Log_HTTP_Complete.

Apêndice B

Scripts

B.1 Robot Framework

```
*** Settings ***
Library  SeleniumLibrary

*** Variables ***
${BROWSER}  chrome
${URL}      https://gibiru.com/
@{Busca_list} tecnologia    educacao    teletrabalho    teleaula    covid
    -19    empregos
*** Test Cases ***
LoginTest
    open browser    ${URL}    ${BROWSER}
    maximize browser window
    acessogibiru
    close browser
*** Keywords ***
acessogibiru
    ${busca}=    Evaluate    random.choice($busca_list)    random
    input text    xpath:/html/body/header/div/div[2]/form/div/input    ${
        busca}
    sleep    3
    Log To Console  palavra escolhida da lista ${busca}
    sleep    3
    press keys    None    ENTER
    sleep    3
```

```
23      ${Page1}=    get element count    xpath=//a[not(contains(@href,'https'))]
        Log To Console    ${Page1}
25      @{linkItens1}    create list
        :FOR    ${i}    IN RANGE    1        ${Page1}
27      \    ${list1}    get text    xpath=(//a[not(contains(@href,'https'))])[${i
            }]
        \    Log To Console      ${list1}
29      sleep    10
```

B.2 Script Xubuntu 18.04

```
1  #!/bin/bash
   curr_date=$(date +'%Y_%m_%d_%H_%M')
3
   sudo tcpdump -i wlp3s0 -w $curr_date-navegador.pcap &
5  sleep 10
   echo teste_bing
7  robot --timestampoutputs --log Navegacao_geral-bing.html --report -r
       ch_navbing.robot > \$curr_date-ch-bing.txt
   echo teste_duckgo
9  robot --timestampoutputs --log Navegacao-duckgo.html --report -r
       ch_navduckgo.robot > $curr_date-ch-duckgo.txt
   echo teste_google
11 robot --timestampoutputs --log Navegacao-google.html --report -r
       ch_navgoogle.robot > $curr_date-ch-google.txt
   echo teste_gibiru
13 robot --timestampoutputs --log Navegacao-gibiru.html --report -r
       ch_navgibiru.robot > $curr_date-ch-gibiru.txt
   echo teste_yahoo
15 robot --timestampoutputs --log Navegacao-yahoo.html  --report -r
       ch_navyahoo1.robot > $curr_date-ch-yahoo.txt
```

B.3 Sript Windows 10

```
@echo off
setlocal ENABLEDELAYEDEXPANSION

set today=!date:/=-!
set now=!time::=-!
set millis=!now:*.=!
set now=!now:.%millis%=!

tcpdump.exe -w !today!_!now!-navagador.pcap
sleep 10
echo teste_bing
robot --timestampoutputs --log navegacao_geral-bing.html --report -r
    ch_navbing.robot > $!today!_!now!-bing.txt
echo teste_duckgo
robot --timestampoutputs --log navegacao-duckgo.html --report -r
    ch_navduckgo.robot > $c!today!_!now!-duckgo.txt
echo teste_google
robot --timestampoutputs --log navegacao-google.html --report -r
    ch_navgoogle.robot > $!today!_!now!-google.txt
echo teste_gibiru
robot --timestampoutputs --log navegacao-gibiru.html --report -r
    ch_navgibiru.robot > $!today!_!now!-gibiru.txt
echo teste_yahoo
robot --timestampoutputs --log navegacao-yahoo.html --report -r ch_navyahoo1
    .robot > $!today!_!now!-yahoo.txt
```

Bibliografia

[1] "Relatório de Cripotografia HTTPS na WEB - JUL/2019". Disponível em https://transparencyreport.google.com/https/overview.

[2] NC Solutions. "Global Encryption - Trends Study". Ponemom Institute, 2018.

[3] Tanenbaum, Andrew S. "Redes de computadores", 5a ed., Rio de Janeiro: Pearson, 2010.

[4] Kurose, James F. e ROSS, Keith W. "Redes de Computadores e a Internet: Uma Abordagem Top-Down"; tradução Arlete Simille Marques; revisão técnica Wagner Luiz Zucchi. 3a ed., São Paulo: Pearson, 2006.

[5] Harrison, M. A., Ruzzo, W. L., e Ullman, J. D. "On protection in operating systems". Proceedings of the 5th ACM Symposium on Operating Systems Principles, (SOSP) 1975.

[6] Pfleeger CP., Pfleeger SL. "Security in computing. 4a ed. Estados Unidos: Prentice Hall; 2007

[7] Kallam, S. "Diffie-Hellman:Key Exchange and Public Key Cryptosystems". Math and Computer Science Department Indiana State University, 2015.

[8] Melo, Sandro. "Exploração de Vulnerabilidades de Redes TCP IP", 3a ed., Rio de Janeiro: Alta Books, 2017.

[9] Dlamini, M. T., Eloff, J. H. P., e Eloff, M. M."Information security: The moving target". Computers and Security - Elsevier, 2009.

[10] Velan, P., Čermák, M., Čeleda, P., e Drašar, M. "A survey of methods for encrypted traffic classification and analysis". International Journal of Network Management, 2015.

[11] Torres, G. "Redes de Computadores: Curso Completo". Rio de Janeiro: Axcel Books, 2001.

[12] Felippetti, M. Aurélio. "CCNA 6.0 Guia Completo de Estudo". Florianópolis: Visual Books, 2016.

[13] Stenberg, D. "HTTP2 Explained. Computer Communication Review". disponível em: https://daniel.haxx.se/http2/.

[14] Satapathy, A., e Livingston, J. "A Comprehensive Survey on SSL/ TLS and their Vulnerabilities". International Journal of Computer Applications, 2016.

[15] Roskind J. "QUIC: Design Document and Specification Rationale". Disponível em: https://goo.gl/eCYF1a.

[16] Stallings, W. "Criptografia e Segurança de Redes de Redes: Princípios e Práticas". tradução Daniel Vieira; revisão técnica Paulo Barreto e Rafael Misoczki. 6a ed., São Paulo: Pearson, 2015.

[17] Elgohary, A., Sobh, T. S., e Zaki, M. "Design of an enhancement for SSL/TLS protocols". Computers and Security - Elsevier, 2006.

[18] Finsterbusch, M., Richter, C., Rocha, E., Müller, J. A., e Hänßgen, K. A survey of payload-based traffic classification approaches. IEEE Communications Surveys and Tutorials, 2014.

[19] Barlet-Ros Co-Advisor, P., e Solé-Pareta, J. "Network Traffic Classification: From Theory to Practice Valentín Carela-Español. Universidade da Catatalunha, 2014.

[20] Karagiannis, T., Papagiannaki, K., e Faloutsos, M. "Blinc: Multilevel traffic classification in the dark," Special Interest Groupon data Communication, (SIGCOMM), 2005.

[21] "Performance Measurement Tools Taxonomy". Disponível em: http://www.caida.org/tools/taxonomy/performance.xml.

[22] Kitchenham, B. e Charters, S. "Guidelines for performing Systematic Literature Reviews in Software Engineering", Engineering, vol. 2. Durham, 2007.

[23] "Infosheet 2016 - Center for Applied Internet Data Analysis". Disponível em: http://www.caida.org/publications/posters/eps/caida-infosheet-2016.pdf.

[24] Amara, S., Macedo, J., Bendella, F., e Santos, A. "Group formation in mobile computer supported collaborative learning contexts: A systematic literature review". Educational Technology and Society, 2016.

[25] Liu, C., Han, J., e Wei, Q. "Browser Identification Based on Encrypted Traffic". International Conference on Communications, Information Management and Network Security, (Cimns), 2016.

[26] Gill, P., e Williamson, C. "Characterizing Organizational Use of Web-based Services: Methodology, Challenges, Observations, and Insights". ACM Journal, 2015.

[27] Kim, S. M., Goo, Y. H., Kim, M. S., Choi, S. G., e Choi, M. J. "A method for service identification of SSL/TLS encrypted traffic with the relation of session ID and Server IP". 17th Asia-Pacific Network Operations and Management Symposium: Managing a Very Connected World, (APNOMS), 2015.

[28] Durumeric, Z., Kasten, J., Bailey, M., e Halderman, J. A. "Analysis of the HTTPS certificate ecosystem". Proceedings of the ACM SIGCOMM Internet Measurement Conference, ACM IMC, 2013.

[29] Muehlstein, J., Zion, Y., Bahumi, M., Kirshenboim, I., Dubin, R., Dvir, A., e Pele, O. "Analyzing HTTPS encrypted traffic to identify user's operating system, browser and application". IEEE Access, 2017.

[30] Fang, C., Liu, J., e Lei, Z. "Fine-Grained HTTP Web Traffic Analysis Based on Large-Scale Mobile Datasets". IEEE Access, 2016.

[31] Manzoor, J., Drago, I., e Sadre, R. "How HTTP/2 is changing web traffic and how to detect it". Proceedings of the 1st Network Traffic Measurement and Analysis Conference, (TMA), 2017.

[32] Wijnants, M., Marx, R., Quax, P., e Lamotte, W. "HTTP/2 Prioritization and its Impact on Web Performance". International World Wide Web Conference Committee - ACM Library, 2018.

[33] Husák, M., Cermák, M., Jirsík, T., e Pavelčeleda, P. P. "HTTPS traffic analysis and client identification using passive SSL/TLS fingerprinting". EURASIP Journal on Information Security, 2016.

[34] Felt, A. P., Barnes, R., King, A., Palmer, C., Bentzel, C., Tabriz, P. "Measuring HTTPS adoption on the web". 26th USENIX Security Symposium, 2017.

[35] Arnbak, A., Asghari, H., Van Eeten, M., e Van Eijk, N. "Security collapse in the HTTPS market". Communications of the ACM, 2014.

[36] Naylor, D., Finamore, A., Leontiadis, I., Grunenberger, Y., Mellia, M., Munafò, M., e Steenkiste, P. "The Cost of the "S" in HTTPS". 10th International Conference on Emerging Networking EXperiments and Technologies - ACM Digital Library, (CoNEXT), 2014.

[37] Gonzalez, R., Soriente, C., e Laoutaris, N. "User profiling in the time of HTTPS". Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, 2016.

[38] Kausar, F., Aljumah, S., Alzaydi, S., e Alroba, R. "Traffic Analysis Attack for Identifying Users' Online Activities". IEEE Computer Society, 2019.

[39] Al Khater, N., e Overill, R. E. "Network traffic classification techniques and challenges". The 10th International Conference on Digital Information Management,(Icdim), 2015.

[40] Finsterbusch, M., Richter, C., Rocha, E., Müller, J. A., e Hänßgen, K. "A survey of payload-based traffic classification approaches". IEEE Communications Surveys and Tutorials, 2014.

[41] Alcock, S., e Nelson, R. "Libprotoident: Traffic Classification Using Lightweight Packet Inspection". WAND - Network Research Group. 2011.

[42] Alcock, S., Lorier, P., e Nelson, R. "Libtrace: A Packet Capture and Analysis Library". International Conference on Parallel and Distributed Computing, Applications and Technologies - PDCAT. Nova Zelândia, 2008.

[43] Mellia, M., Carpani, A., e Cigno, R. Lo. "TStat: TCP STatistic and Analysis Tool". LNCS (Vol. 2601). Torino, 2003.

[44] Lawrence, E. "An Overview of Telerik Fiddler". Disponível em https://www.telerik.com/blogs/an-overview-of-telerik-fiddler.

[45] "Pcapplusplus - Feature Overview", Disponível em https://pcapplusplus.github.io/docs/features.

[46] "Robot Framework for Web Tests". Disponível em https://robotframework.org.

[47] "Selenium Library - Python". Disponível em https://selenium-python.readthedocs.io.

[48] "Pycharm - IDE". Dispovível em https://www.jetbrains.com/pycharm/.

[49] Grikorik, Y. "High Performance - Browser Networking". Estados Unidos: O'Reilly, 2013.

[50] "Chrome e outros navegadores vão abandonar protocolo usado há quase 20 anos"Disponível em https://olhardigital.com.br/noticia/chrome-e-outros-navegadores-vao-abandonar-protocolo-usado-ha-quase-20-anos/79202.