INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE RONDÔNIA
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
TRABALHO DE CONCLUSÃO DE CURSO

WIRESHARK - Detecção de pacotes/conexões HTTP vulneráveis em uma rede sem fio

FRANCIELE GOMES DE MOURA

Ji-Paraná, 2019

DESENVOLVIMENTO DE PRODUTO CIENTÍFICO E TECNOLÓGICO

Wireshark - Detecção de pacotes/conexões HTTP vulneráveis em uma rede sem fio

Franciele Gomes de Moura

Trabalho de Conclusão de Curso apresentado ao curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, como parte dos requisitos necessários à obtenção do título de tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: José Lucas Brandão Montes

Ji-Paraná, 2019

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO, com dados informados pelo(a) próprio(a) autor(a).

Moura, Franciele Gomes de.
WIRESHARK - Detecção de pacotes/conexões HTTP vulneráveis em uma rede sem fio / Franciele Gomes de Moura, Ji-Paraná-RO, 2024.
51 f. : il.

Orientador(a): Prof. José Lucas Brandão Montes.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Ji-Paraná-RO, 2024.

1. Wireshark. 2. Http. 3. Vulnerabilidade. I. Montes, José Lucas Brandão (orient.). II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

Bibliotecário(a) Responsável: Cleuza Diogo Antunes, CRB-11/864 (Campus Ji-Paraná)

Franciele Gomes de Moura

WIRESHARK - Detecção de pacotes/conexões HTTP vulneráveis em uma rede sem fio

Trabalho de Conclusão de Curso apresentado ao curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, como parte dos requisitos necessários à obtenção do título de tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: José Lucas Brandão Montes

Aprovado pela Banca Examinadora em 11 de novembro de 2019.

BANCA EXAMINADORA

_____

Orientador:José Lucas Brandão Montes

_____

Professor: Clayton Ferraz Andrade

_____

Professor: Walter Ferreira Siqueira

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus por me levantar todos os dias, familiares e amigos que sempre me incentivaram.

Minha mãe pelas “lindas” palavras de entusiasmo.

Meu pai por me levar várias vezes de moto até a faculdade, porque o ônibus não ia.

AGRADECIMENTOS

Primeiramente, e acima de tudo, agradecer a Deus pelo dom da vida e da sabedoria para conseguir finalizar este trabalho.

Segundo, a minha família que, de modo diferente, me ajudaram a alcançar meus objetivos e não desistir de terminar a faculdade, por mais difícil e cansativa que tenha sido.

Terceiro, e não menos importante, aos professores do curso de Análise e Desenvolvimento de Sistemas, que conseguiu conceito máximo no MEC (Ministério da Educação) - muito merecido, na minha opinião -, por passarem um pouco do seu vasto conhecimento a nós.

"Se você não gosta do seu destino, não aceite. Em vez disso, tenha a coragem de o mudar do jeito que você quer que seja"

(Uzumaki, Naruto)

RESUMO

O presente trabalho propõe uma solução para o problema enfrentado que provém da falta de sigilo em transações que ocorrem no protocolo HTTP. Uma vez que, pessoas mal intencionadas utilizam a ferramenta *wireshark* para capturar pacotes com protocolo HTTP que podem conter informações sigilosas de terceiros (como dados de *login* - usuário e senha de *sites*, aplicativos, entre outros), pelo simples fato de estar conectado na mesma rede que este último, sendo assim, será utilizado uma ferramenta de implementação de rede privada virtual para omitir informações sigilosas através da rede. Tal ferramenta possui a capacidade de criptografar dados que serão enviados pela rede, garantindo que eles cheguem ao destinatário sem que sejam capturados ou lidos por terceiros que não tenham permissão.

Palavras-chave: wireshark; http; vulnerabilidade.

ABSTRACT

This paper proposes a solution to the problem that arises from the lack of confidentiality in transactions that occur in the HTTP protocol. Because malicious people use the wireshark tool to capture HTTP protocol packets that may contain sensitive third-party information (such as username and password for websites, applications, and so on) simply because they are connected to it. The latter will therefore use a virtual private network implementation tool to omit sensitive information across the network. Such a tool has the ability to encrypt data that will be sent over the network, ensuring that it reaches the recipient without being captured or read by unauthorized parties.

Keywords: wireshark; http; vulnerability.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface (Interface de Programação de Aplicativos).

ARP - Address Resolution Protocol (Protocolo de resolução de Endereços).

Bits - Binary Digit (Dígito Binário).

CERT.br - Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil.

DHCP - Dynamic Host Configuration Protocol (Protocolo de configuração dinâmica de hosts).

EUA - Estados Unidos da América.

FTP - File Transfer Protocol (Protocolo de Transferência de Arquivos).

GPL/GNU - General Public License (Licença Pública Geral GNU).

GTK (GIMP) - Toolkit multiplataforma.

HTML - Hypertext Markup Language (Linguagem de Marcação de Hipertexto).

HTTP - HyperText Transfer Protocol (Protocolo de Transferência de Hipertexto).

HTTPS - HyperText Transfer Protocol Secure (Protocolo de Transferência de Hipertexto Seguro).

IBGE - Instituto Brasileiro de Geografia e Estatística.

IP - Internet Protocol (Protocolo de Internet).

LibPCap - Biblioteca para as versões Unix.

MAC - Media Access Control (Controle de Acesso a Mídia).

Megabits - Unidade de transmissão de dados.

NIC.br - Núcleo de Informação e Coordenação do Ponto BR.

PARC - Empresa Xerox Palo Research Center, Palo Alto, Califórnia, nos Estados Unidos.

PCap - Biblioteca Packet Capture (PCap) para capturar o tráfego de rede.

QT - framework voltado para o desenvolvimento de interfaces gráficas.

RARP - Reverse Address Resolution Protocol (Protocolo de Resolução Reversa de Endereço).

SSH - Secure Shell (Cápsula Segura).

SSL - Secure Sockets Layer (Camadas Seguras de Soquetes).

TCP - Transmission Control Protocol (Protocolo de Controle de Transmissão).

Telnet - Protocolo de rede.

TSL - Transport Layer Security (Segurança da Camada de Transporte).

UDP - User Datagram Protocol (Protocolo de Datagrama do Usuário).

VPN - Virtual Private Network (Rede Virtual Privada).

WinPCap - Biblioteca PCap para as versões Windows.

WEB - Sistema hipertextual que opera através da *internet*.

WWW - World Wide Web.

SUMÁRIO

1 INTRODUÇÃO

O presente trabalho trouxe a problemática que envolve o mundo digital, uma vez que, mediante simples ato de se conectar a uma determinada rede, várias informações do usuário são cedidas, para servir de identificação. Mediante isto, a quantidade cada vez maior de aplicações, serviços e protocolos de aplicação dificultam a tarefa de manter a rede parcialmente segura, pois existem vulnerabilidades, riscos e muitas ameaças de invasão de uma rede. Por isso, surgiu a necessidade de fazer um gerenciamento eficiente de rede, visando manter a segurança no tráfego que passa por ela.

Realizar uma análise constante da rede local, é uma forma de gerenciar, e é importante, uma vez que, a partir disso percebe-se ataques ou problemas que precisam ser resolvidos. Além de possibilitar ver dados de *login* - usuário e senha -, caso algum *site* não proporciona a transferência criptografada dos dados. Está última funcionalidade é bem desfrutado por pessoas mal intencionadas que pretendem roubar informações de terceiros. Para que os pacotes que transportam informações sigilosas não passem em texto claro pela rede a utilização de uma rede virtual pode ser uma solução, visto que, garante sigilo nas transações, pois criptografa os dados, colocando-os dentro de outro pacote e os enviando por um túnel até o destinatário. Mesmo que estes pacotes foram capturados, seus dados não serão lidos.

1.1 Justificativa

No atual mundo digital, está conectado se tornou uma necessidade básica, uma vez que, 74,9% da população brasileira em 2017 utiliza a *internet*[1] (IBGE, 2018). Quando não suprida essa necessidade, causa desconforto nas pessoas devido a importância inevitável de ler e-mails, enviar ou acessar dados do trabalho, realizar pesquisas ou até mesmo se orientar através de um mapa para encontrar determinado lugar.

Ao conectar a rede, é necessário ceder algumas informações, que servem como identificação ou registro, de que um usuário, programa ou processo realizou determinada atividade. Assim sendo, a preocupação com a segurança se torna um fator muito importante, uma vez que, as informações estão expostas a uma grande variedade de ameaças e vulnerabilidades. Seja por ameaça de vírus ou de *crackers*[2].

---

[1] *Internet* - conjunto de rede mundial de computadores.

[2] *crackers* (quebrado) - pessoas mal intencionadas que praticam atos ilegais e antiéticos;

Por isso, vários métodos foram criados para melhorar a proteção das informações que trafegam por uma rede. Mantendo a confidencialidade - garantir que o acesso à informação seja limitada apenas às pessoas que possuem autorização; a integridade - garantir que as informações originais não foram alteradas; e a disponibilidade - garante que a informação esteja sempre disponível para os usuários autorizados (CERT.br, 2017).

Muitas ferramentas e *softwares*[3] surgiram para auxiliar os usuários e os profissionais de rede a fazerem o monitoramento e gerenciamento das informações que trafegam por ela. Um exemplo é a ferramenta *wireshark*[4], um *sniffer*[5] de rede, que captura e armazena os pacotes que entram e saem de uma rede de computador (CERT.br, 2017).

O *wireshark* tem uma interface gráfica que possibilita uma análise minuciosa dos dados, pois fica visível as informações de um pacote capturado, como o número, o tempo que levou para capturar, o endereço de origem e de destino, o nome do protocolo e informações adicionais (podendo constar nomes de usuários e senhas). Por ser uma ferramenta gratuita e de fácil utilização, muitas pessoas mal intencionadas a utilizam para roubar informações confidenciais de terceiros, pelo simples fato de estarem conectados à mesma rede (SNIFF FREE OR DIE, 2018).

Partindo do cenário descrito anteriormente, a utilização da ferramenta *wireshark*, para detectar conexões vulneráveis em uma rede sem fio de computador em um ambiente computacional, onde o tráfego pode ser intenso, é de suma importância. Por isso, o foco do projeto é a utilização de ferramentas que possibilitem que o usuário não tenha seus dados pessoais trafegando por pacotes HTTP vulneráveis na rede em texto claro[6], aumentando a segurança quando navegar pela *internet*. Considerando o pressuposto apresentado, é possível utilizar, em uma rede vulnerável e monitorada, alguma ferramenta que possibilita ocultar os dados trafegados por essa rede?

Mediante a problemática levantada no presente trabalho, acredita-se que, com a utilização de ferramentas de VPN é possível ocultar as mensagens trocadas entre dois pontos em uma rede, fazendo com que, mesmo que as mensagens sejam capturadas, elas não serão lidas, fato que será verificado e atestado durante a construção do presente trabalho (NAKAMURA, 2007).

---

[3] *software* - componentes lógicos de um computador ou sistemas;
[4] *wireshark* (tubarão) - analisador de pacotes;
[5] *sniffer* (farejador) - ferramentas que interceptam e analisam o tráfego de uma rede.
[6] Texto claro - informação legível (original, da forma que o usuário digitou).

1.2 Objetivos

Os objetivos do presente trabalho estão delimitados abaixo, divididos em objetivo geral e objetivos específicos.

1.2.1 Objetivo Geral

Demonstrar ferramenta que previne que o software *wireshark* não possa capturar dados confidenciais que trafegam na rede no protocolo HTTP em um ambiente computacional.

1.2.2 Objetivos Específicos

- Abordar conceitos básicos sobre redes, protocolos e *sniffer* pertinentes para o desenvolvimento do projeto;
- Verificar se é possível obter informações confidenciais ao capturar dados pela ferramenta *wireshark*;
- Mostrar ferramenta que possibilita ocultar informações sigilosas.

1.3 Estrutura do Trabalho

O trabalho está dividido em capítulos:

O capítulo 1 especificar conceitos variados sobre redes de computadores, protocolos, *sniffer* entre outros. Tópicos que precisam ser definidos para que o leitor tenha um melhor entendimento no desenvolver do trabalho;

O capítulo 2 traz uma introdução à ferramenta *wireshark*, que é capaz de capturar pacotes de dados que trafegam pela rede, é um *software* gratuito, de código livre, programada nas linguagens C e C++. Utilizada para capturar e analisar pacotes de dados que entram e saem de uma determinada rede;

O capítulo 3 traz a introdução a ferramenta VPN, que possibilita a criptografia, o tunelamento e o encapsulamento dos dados que serão enviados, visando assim, que a mensagem chegue até o destinatário sem sofrer algum tipo de dano ou modificação;

O capítulo 4 analisa uma rede sem fio em um ambiente computacional, onde se tem dados capturados, como dados de *login* - usuário e senha - que podem ser roubados por pessoas mal intencionadas que podem utilizar essas informações de modo antiético. Ainda no quarto capítulo, a utilização da ferramenta VPN nos testes, assim ajuda os usuários não têm suas informações confidenciais passando pela rede em texto claro, dificultando que pessoas mal intencionadas vejam esses dados;

No capítulo 5, os resultados e conclusões chegadas após a realização dos vários testes no ambiente computacional. Demonstrando como o uso da VPN pode ajudar a ocultar informações para que não passem em texto claro pela rede.

2 REFERENCIAL TEÓRICO

Considerando a natureza, objetivos e problemáticas levantados durante a construção do presente trabalho, é necessário abordar conceitos referentes à redes de computadores, tráfego de dados, ferramentas de *sniffer*, entre outros, a fim de embasar teoricamente o que está sendo proposto pelo trabalho.

2.1 Introdução a rede de computadores

Redes de computadores são um conjunto de máquinas ligadas entre si. Foram criadas na década de 60, com o objetivo de transferir informações de um computador para outro. Naquela época, usavam cartões perfurados para armazenar informações externamente e também transferir dados. Esses cartões se pareciam muito com cartão de cartolina com furos, que representam os *bits*[7] 1 e 0. As transferências por cartão perfurado eram trabalhosas, lentas e demoradas, uma vez que, só permitia armazenar em torno de 80 caracteres por cartão (MONQUEIRO, 2008).

Por volta de 1969 a 1972 a *Arpanet*[8], uma rede de computadores, foi criada. Entrou no ar em dezembro de 1969. Seu propósito inicial era de interligar 4 computadores de arquiteturas diferentes. Após, aproximadamente 4 anos, já interligava 30 instituições, dentre elas: universidades, instituições militares e escolas (MONQUEIRO, 2008).

No ano seguinte, 1974, surgiu o TCP/IP, que se tornou o protocolo de uso definitivo da *Arpanet* e mais tarde da *Internet* também. Segundo Monqueiro (2008), "Uma rede interligando universidades permitiu o livre tráfego de informações, levando ao desenvolvimento de recursos que usamos até hoje, como o e-mail, o telnet e o FTP [...]." Isso permitiu que os usuários conectados pudessem acessar outros computadores de forma remota, trocar informações e compartilhar arquivos.

"[...] em 1973 dentro do PARC (o laboratório de desenvolvimento da Xerox, em Palo Alto, EUA), [...] foi feito o primeiro teste de transmissão de dados usando o padrão Ethernet." (MONQUEIRO, 2008). Foi graças a este teste que o padrão *Ethernet*, que transmitia dados a 2.94 *megabits*[9] por meio de cabos coaxiais[10] e permitia a conexão de até 256 estações de trabalho[11], teve sua origem.

---

[7] *Bits* - dígitos binários, a menor unidade de informação que pode ser armazenada e transferida.
[8] *Arpanet* - conhecida como rede-mãe da *internet.*
[9] *Megabits* - unidade de medida.
[10] Cabos coaxiais - cabo condutor feito com cobre que transmite sinais.
[11] Estação de trabalho - computadores situados com potência alta, como um escritório.

O termo "*ether*" era usado para descrever o meio de transmissão dos sinais em um sistema. O padrão *Ethernet* surgiu da necessidade natural de ligar as estações de trabalho em rede. A criação, respectiva, da *Arpanet* e do padrão *Ethernet*, deram origem a *internet* e às redes locais. A abertura de acesso a *internet* só ocorreu na década de 1990 e, a partir daí, as redes se popularizaram de forma assustadora, crescendo, se desenvolvendo, até chegar ao nível que está hoje (MONQUEIRO, 2008).

Para que as redes de computadores possam se comunicar, enviar e receber informações, é necessário a utilização dos protocolos de rede.

2.2 Protocolo de rede

O conceito de protocolo, segundo Torres (2004, p.34), é a "linguagem" usada pelos dispositivos de uma rede de modo que eles consigam se entender, isto é, trocar informações entre si. Seguindo um conjunto de regras que ajuda na administração da comunicação de dados.

A troca de informações entre dispositivos e computadores é chamado de comunicação de dados. Uma vez que, transmite informações como texto, figuras, números, vídeos e áudios entre computadores, estação de trabalho, telefone, câmera, entre outros dispositivos.

Segundo Cantú (2003, p.6), "Os protocolos definem o formato e a ordem das mensagens enviadas e recebidas pelas entidades da rede bem como as ações que são tomadas quando da transmissão ou recepção de mensagens." Ou seja, são normas e padrões que precisam ser seguidos para uma melhor interação na rede. Dentre os muitos protocolos existentes, alguns são:

- **IP** (Protocolo de *Internet*) - é um protocolo de comunicação usado entre as máquinas em rede para encaminhamento de datagramas[12], ou seja, pacotes de dados transmitidos na rede através de um serviço sem conexão (ligação entre dois ou mais sistemas informáticos) (KAMIENSKI et al, 2000, p.12).
- **TCP** (Protocolo de Controle de Transmissão) - oferece um serviço confiável entre aplicações:
    - Permite que duas máquinas controle o estado de transmissão - transmitir informações;
    - Permite a entrega ordenada dos datagramas vinda do protocolo IP;

[12] Datagramas - unidade básica de dados que contém toda a informação necessária que identifica seu conteúdo, além de dados do endereço da origem e do destino.

    - Verifica a onda de dados para evitar uma saturação da rede;
    - Permite formatar os dados em segmentos a fim de “entregar” ao protocolo IP;
    - Fazer circular informações que vem de fontes como aplicações de forma simultânea distintas numa mesma linha;
    - Permite o começo e o fim de uma comunicação de maneira educada e confiável, além de avisar ao cliente e ao servidor a recepção correta e mútua dos dados (BARGE, 2012, p.36-38).
- **UDP** (Protocolo de Datagrama do Usuário) - é parecido com o TCP, só que mais simples. Infelizmente não há procedimento de verificação de recebimento ou envio de dados, e se, por acaso, algum pacote não for recebi, o computador destinatário não faz uma nova solicitação, como ocorre com o TCP. Por isso, o UDP é um pouco mais rápido (DANTAS, 2012, p.153).
- **ARP** (Protocolo de Resolução de Endereços) - permite conhecer (converter) o endereço físico de uma placa de rede, tornando-o correspondente a um endereço IP (CANTÚ, 2003, p.54).
- **RARP** (Protocolo de Resolução Reversa de Endereços) - permite conhecer o endereço IP a partir de uma associação do endereço MAC - é um endereço físico único associado à interface de comunicação, que conecta um dispositivo à rede (CANTÚ, 2003, p.74).
- **FTP** (Protocolo de Transferência de Arquivos) - fornece o serviço de transferência de arquivos. Define a maneira segundo a qual os dados devem ser transferidos numa rede TCP/IP (DANTAS, 2012, p.221).
- **Telnet** - Implementa o serviço de terminal remoto. Permite emular (traduzir/adaptar) um terminal à distância, isto é, permite executar comandos escritos no teclado de um computador remoto. Utiliza o conceito de terminal virtual de rede (DANTAS, 2012, p.218).
- **SSH** (Cápsula Segura) - parecido com o *Telnet* -, permite acessar remotamente outros sistemas, de forma mais segura, pois criptografa as informações na transferência (CANTÚ, 2003, p.74).
- **DHCP** (Protocolo de Configuração Dinâmica de Hosts) - protocolo permite a alocação (atribuição) dinâmica de endereço IP (CANTÚ, 2003, p.71).
- **HTTP** (Protocolo de Transferência de Hipertexto) - protocolo para informações distribuídas, ou seja, comunicação entre sistemas de informação

que permite a transferência de dados entre redes de computadores (FIELDING, 1999, p.1).

- **HTTPS** (Protocolo de Transferência de Hipertexto Seguro) - combinação do HTTP e SSL (Camadas Seguras de Soquetes). Transmite dados entre redes de computadores na *internet* de forma segura, uma vez que, criptografa os dados (ARNBAK, 2014).
- **SSL** (Camadas Seguras de Soquetes) - desenvolvido para garantir a segurança entre aplicações cliente/servidor[13] evitando falsificação ou roubo dos dados e "escutas". Ao ser padronizado recebeu o nome de *Transport Layer Security* (TSL). O TSL 1.0 e o mesmo que o SSL 3.0 (BORGES, 2014, p.8).

Os protocolos ajudam na administração, controlando e regulando a transferência de dados que ocorrem entre sistemas computacionais.

Existe também o modelo de referências **TCP/IP** - (Protocolo de Controle de Transmissão/Protocolo de *Internet*). É o modelo conceitual de rede que reúne um conjunto de protocolos. Suponhamos que protocolo seja uma linguagem. Para que duas pessoas possam se comunicar, é preciso usar a mesma linguagem, um exemplo seria o inglês. Neste caso, o TCP/IP seria a linguagem mundial que será usada por computadores e dispositivos para que possam se comunicar e compartilhar informações entre si.

O TCP/IP surgiu para que os protocolos pudessem se comunicar, por isso, as camadas foram criadas, assim os protocolos das camadas interagem com os outros protocolos (TANENBAUM, 1997). Vários autores defendem que o TCP/IP possui 5 camadas, neste trabalho usaremos como referências o autor Tanenbaum, que trás apenas quatro:

- Camada de Aplicação - responsável pela comunicação entre as aplicações. Ex.: Telnet, FTP, HTTP, etc;
- Camada de Transporte - permite o encaminhamento de dados entre a origem e o destino. Ex.: TCP e UDP;
- Camada de Internet - permite a transferência de pacotes em qualquer rede. Ex.: IP, ARP, etc;
- Camada de Acesso à rede - específica como os dados devem ser encaminhados. Ex.: *Arpanet* (TANENBAUM, 1997).

[13] Aplicações cliente/servidor - um processo onde os servidores são responsáveis pela manutenção da informação e os clientes pela obtenção dos dados.

Fonte: Conteúdos do curso de SI[14]

As quatro camadas, com seus respectivos protocolos, possuem a responsabilidade de trafegar a informação pela rede entre aplicações, fazendo que a informação passe pelas camadas e chegue ao usuário.

Os protocolos passam pela rede de computadores, levando e trazendo informações, podem ser capturados por uma ferramenta *sniffer*, um *software* farejador .

2.3 Sniffer

A quantidade cada vez maior de aplicações, serviços e protocolos de aplicação dificultam a tarefa de manter a rede parcialmente segura, pois existem vulnerabilidades, riscos e muitas ameaças de invasão contra uma rede. De modo que, ferramentas de *sniffers* proporcionam o objetivo de analisar sua rede, monitorando o tráfego, alertando sobre possíveis problemas.

Sniffer é um *software* que captura o tráfego que passa por uma rede, entrando ou saindo. Existem diversos tipos de *sniffers* como: de pacotes, *wi-fi*, redes, IP, entre outros. São conhecidos também como analisador de protocolos, pacotes ou de redes. É um *software* que tem a função de interceptar e registrar os dados que estejam trafegando pela rede. Comumente utilizado por administradores de rede, para analisar e monitorar o tráfego da mesma, para que

[14] Conteúdos do curso de SI - https://marrciohenrique.wordpress.com/2014/03/22/modelo-tcpip-e-osi/.

assim possam detectar problemas existentes. Outras vezes, usados por *crackers*[15] para terem acesso a informações confidenciais de terceiros (HORA, 1997).

Sniffers são responsáveis por "capturar o tráfego da rede interna, podem revelar senhas, esquemas de autenticação de serviços, além de dados processados e enviados por aplicações que necessitam de sigilo na transação." (HORA, 1999, p.2). O que pode resultar em roubo de informações por pessoas mal intencionadas que utilizam da fragilidade da defesa para apropriar-se de dados de forma ilegal.

A ferramenta *sniffer, Wireshark,* é uma das mais utilizadas no mundo, de acordo com o *site Sniff Free or Die*, publicação de 2018. É de código livre[16] e pode ser baixada por várias plataformas operacionais. Por isso, e pelo fato de possuir uma interface gráfica[17], que o diferencia dos outros analisadores de protocolos, foi o escolhido para ser utilizado no decorrer do trabalho. Uma vez que, mediante a interface gráfica o entendimento sobre redes de computadores, protocolos, pacotes e outros conceitos não precisa ser tão abrangente para se obter as informações que se precisa a partir da captura dos dados pela ferramenta. Mas as outras ferramentas *sniffer* necessitam de um conhecimento mais abrangente para encontrar as informações que são disponibilizada em códigos de linhas, onde ficam dispostas informações complicadas que só conhecedores de rede poderiam compreender mais facilmente.

2.4 Wireshark

O surgimento da ferramenta *wireshark*[18] - um analisador de protocolos de rede, que concebe a captura de dados -, data de mais de 20 anos atrás. Quando Gerald Combs estudava ciência da computação na *University of Missouri,* que fica na cidade do *Kansas,* e trabalhava em um pequeno provedor de *Internet*. Nesses provedores era muito comum darem problemas, por isso, Combs procurava ferramentas que ajudasse no amparo a resolução deles. Mas ferramentas que faziam a análise de protocolos de redes daquela época não funcionava no *Linux* e *Solaris,* sistemas operacionais[19] da empresa que ele trabalhava. Além de que, eram caros, custavam em torno de $1.500. Foi por isso que Gerald iniciou seu próprio analisador de tráfego de rede, tendo o propósito de poder rastrear os problemas que ocorriam nos sistemas (INFRA, 2010).

---

[15] *Crackers* - pessoas mal intencionadas que comete atos ilegais.
[16] Código livre - código aberto.
[17] Interface gráfica - ilustração gráfica de uma tela de programa de computador.
[18] *Wireshark* - tubarão de fios.
[19] Sistemas operacionais - programas iniciam o *hardware* do computador,permitindo a interação entre usuário e máquina.

A primeira versão da ferramenta foi lançada em 1998, com o nome de *Ethereal* 0.2.0, e teve trinta atualizações só no primeiro ano, pois os serviços de hospedagem de produtos de código aberto não existiam naquela época.

No ano de 2006, no mês de maio, Gerald Combs começou a trabalhar para a *CACE Technology*. Nesta época, metade do seu código fonte do *Ethereal* foi distribuído pela licença GNU GPL, a outra metade continuou pertencendo a Combs, que usou como repositório para o futuro *wireshark*. A *Riverbed Technology* comprou, em 2010, a *CACE* e passou a ser o patrocinador principal da ferramenta e também do evento *Shark Fest* - conferências anuais educacionais que acontecem acerca do *wireshark* (INFRA, 2010).

Como a *Network Integration Services* possui o domínio da marca *Ethereal*, Combs teve que mudar o nome da sua ferramenta. Passando assim, a se chamar de *Wireshark*. Uma notificação foi enviada para os antigos clientes do *Ethereal*, recomendando mudar para o *wireshark* (INFRA, 2010).

Esta ferramenta captura dados e é uma das mais utilizadas do mundo, segundo *Sniff Free or Die* (2018), mencionado anteriormente. É um *software* gratuito, de código livre, programada nas linguagens C e C++. Sua última versão 3.0.4 foi lançada em 11 de setembro de 2019, tem suporte para as plataformas: *Linux, Solaris, Windows, FreeBSD, OpenBSD, NetBSD* e *MAC OS*. Podendo ser baixado pela plataforma oficial da ferramenta, disponível em: https://www.wireshark.org/.

Utiliza o GTK (GIMP - Toolkit), que é um toolkit - elemento de interação, como janelas, menus, ícones, etc - usado na criação de interfaces gráficas. É um software multi-plataforma que utiliza o GTK (GIMP - GNU Image Manipulation Program - Tool Kit) para implementar a sua interface gráfica com o usuário, e utiliza a biblioteca PCap para capturar pacotes que trafegam pela rede (FARRUCA, 2009, p.14).

A biblioteca *Packet Capture* (PCap) é *open source*[20], que disponibiliza uma interface de programação (API) para quase todos os sistemas operacionais e é a base das ferramentas que fazem a captura de pacotes recebidos ou enviados por uma interface de rede de computador (FARRUCA, 2009, p.17-18). API, pode fazer a captura de pacotes da rede e possibilita salvá-los em ficheiros (arquivo), desta forma, a análise pode ser feita posteriormente. A biblioteca PCap recebe o nome de *WinPCap* para as versões do *windows*, e *LibPCap* para as versões *Unix*.

[20] Open source - código livre.

A interface utilizada pelo *wireshark* é o Qt[21]. *Framework* é um conjunto de conceitos usados para resolver um problema. Qt é um *framework* multiplataforma para interface gráficas em C++. Este *framework*[22] possibilita programar bibliotecas e aplicativos de uma só vez, e ao compila-los, eles rodem em diversas plataformas sem precisar modificar o código fonte (SNIFF FREE OR DIE, 2018).

Para se fazer a captura dos pacotes que estão passando pela rede, a ferramenta utiliza a placa de rede em modo promíscuo. Placa de rede, segundo Meirelles (2002), "é o hardware que permite aos micros conversarem entre si através da rede. Sua função é controlar todo o envio e recebimento de dados através da rede". Ou seja, cada computador terá sua placa de rede, e só é possível acessar a rede por meio dela.

Ao usar a placa de rede em modo promíscuo, além de poder conectar-se na rede, também possibilita que a ferramenta *wireshark* possa capturar tudo independente para qual seja o endereço de destino que determinado pacote tenha sido enviado. Além de proporcionar o armazenamento destes para análise (DA SILVA, 2010).

Dentre outros recursos da ferramenta, podemos citar cinco preferências principais:

- **Interface com o usuário** - determina como o *wireshark* apresenta os dados. Pode-se alterar a maioria das opções de acordo com as preferências de cada usuário, incluindo salvar ou não as posições da janela, o layout dos três painéis principais, a colocação da barra de rolagem, a colocação da lista de pacotes, as colunas dos painéis, as fontes usadas para exibir os dados capturados, e o fundo e as cores de primeiro plano;
- **Opções de captura** - pode usar filtros, visualizando assim, só o que for mais relevante;
- **Impressão** - permite que se faça a impressão dos dados capturados;
- **Resolução de nomes** - permite resolver endereços para nomes mais reconhecidos, incluindo endereços MAC;
- **Protocolos** - permite a manipulação da captura e exibição dos protocolos que o *wireshark* é capaz de decodificar. Estas opções são deixadas inalteradas a menos que o usuário tenha um motivo específico para fazê-las (SANDERS, 2010, p.29).

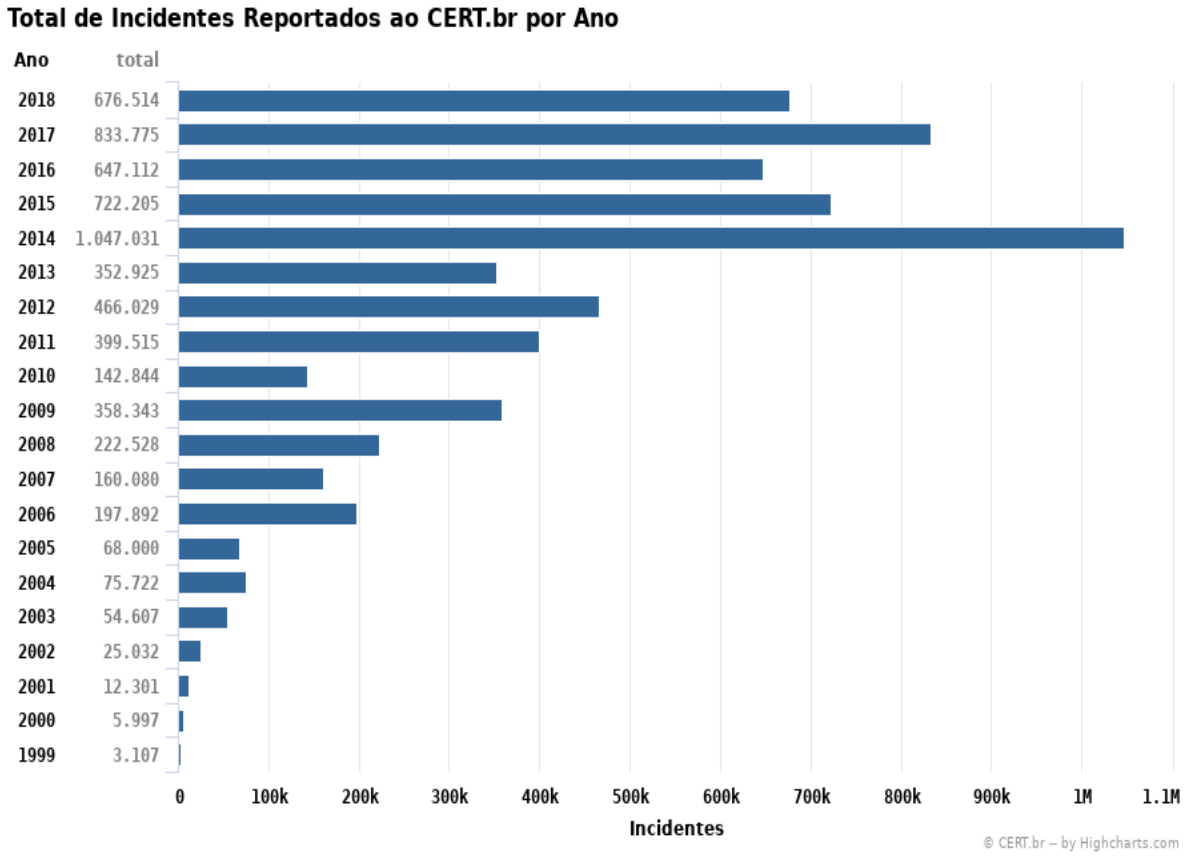
Para um melhor entendimento do uso da ferramenta nos testes que se seguiram, o próximo tópico é uma breve introdução de algumas funções interessantes que a ferramenta traz.

[21] Qt - cute, em inglês.
[22] *Framework* - códigos comuns que fornecem funcionalidades globais para projeto de *software*.

2.4.1 Introdução ao wireshark

Segundo estatísticas de incidentes reportados ao CERT.br - Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil, mantido pelo NIC.br, do Comitê Gestor da Internet no Brasil, e atende a qualquer rede brasileira conectada à Internet - (2019):

Figura 1 - Total de Incidentes Reportados ao CERT.br



Fonte: CERT.br (2019).

De acordo com os dados da pesquisa, mais de 676 mil usuários sofreram algum tipo de incidente na sua rede ano passado (2018). Esses acontecimentos são decorrentes de *vírus*, desfiguração de página[23], força bruta[24], *sniffing*[25], *scan*[26], entre outros (CERT.br, 2019).

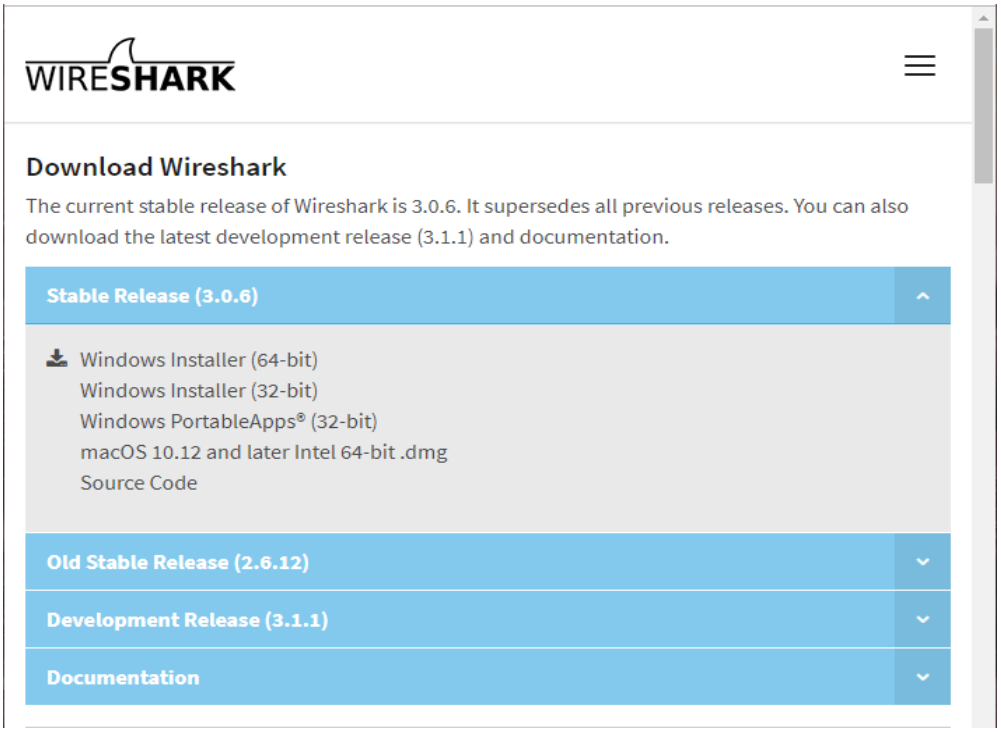
[23] Desfiguração de página - o conteúdo da página web de um site sofre alteração;
[24] Força bruta - ataque que tenta adivinhar o nome do usuário e senha de sites a partir de tentativa e erro;
[25] Sniffer - captura de tráfego;
[26] Scan - busca minuciosa na rede para coletar informações.

Neste trabalho, será mostrado como o *sniffing* funciona, uma das técnicas que um *cracker* faz para obter informações em uma rede *WI-FI* por meio do protocolo HTTP. Por meio da prática de *sniffing* é capaz de decodificar o conteúdo dos pacotes trocados entre dispositivos de uma rede.

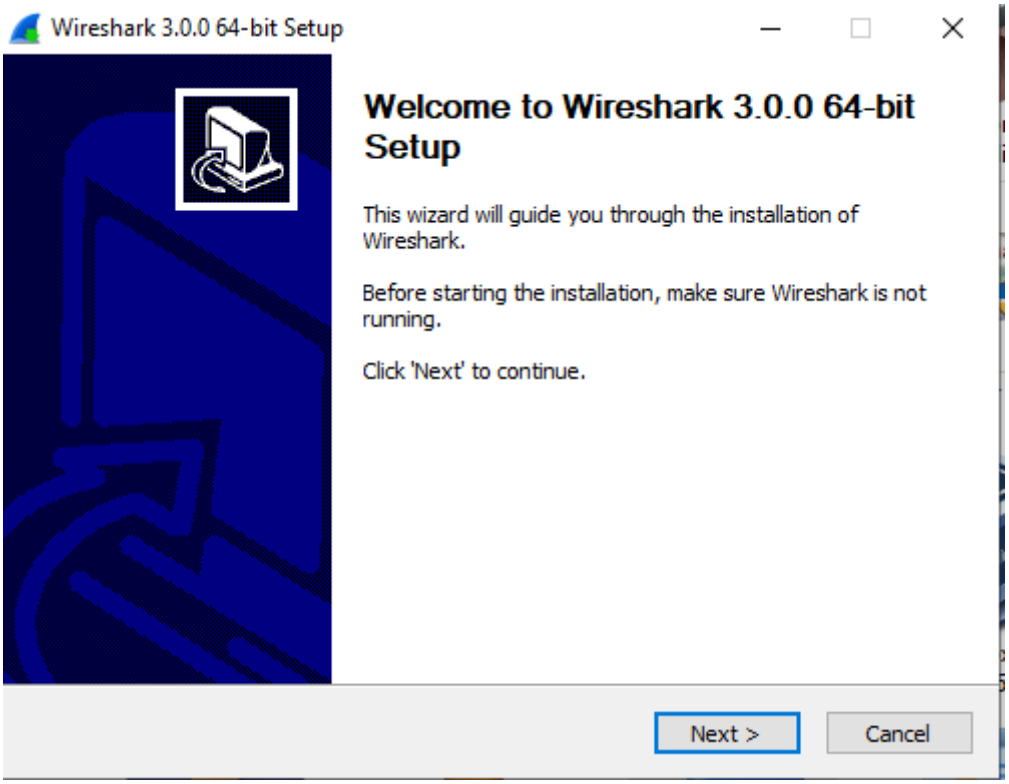
A rede *WI-FI* é um padrão usado para se comunicar, de forma sem fio em banda larga, ou seja, a conexão a *internet* é feita acima da velocidade padrão. É utilizado para fazer conexões entre computadores ou dispositivo (LEE; SU; SHEN, 2007).

O uso do *wireshark* é bem simples, desde a instalação ao manuseio, mas precisa-se de um conhecimento prévio sobre redes e protocolos para identificar os pacotes que são mais importantes de serem analisados.

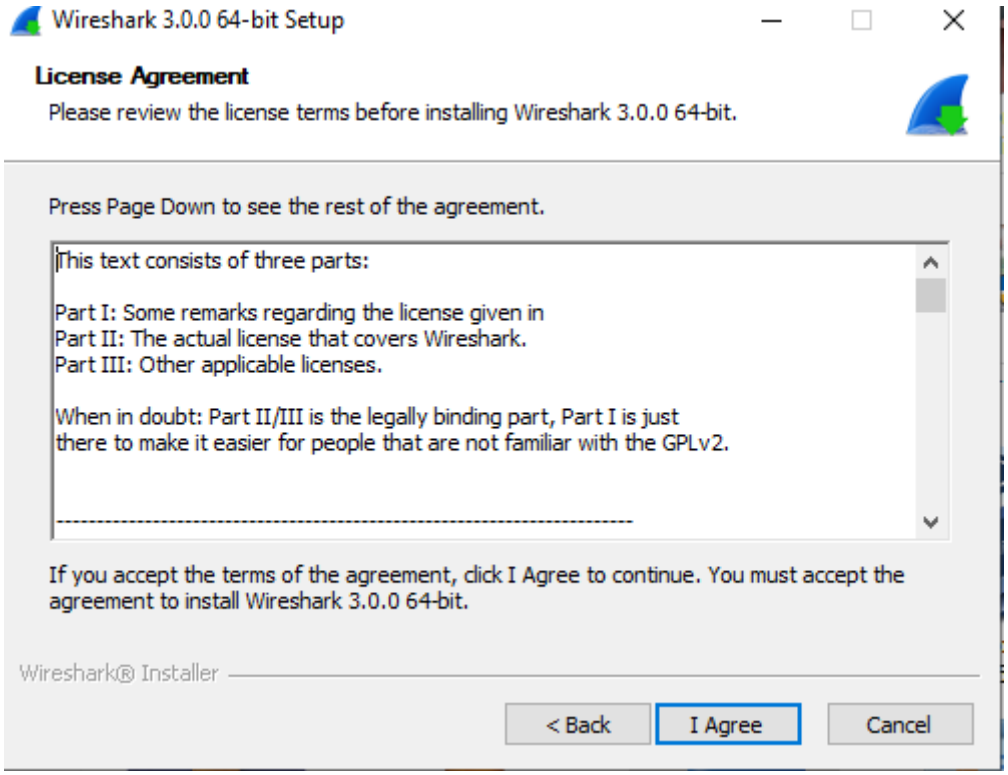
A página de *download* da ferramenta está disponível em: https://www.wireshark.org/, *site* oficial, onde pode-se escolher o sistema operacional e a versão que pretende baixar, sendo de 32 ou 62 *bits*, no caso do *windows*, como na imagem a baixo.



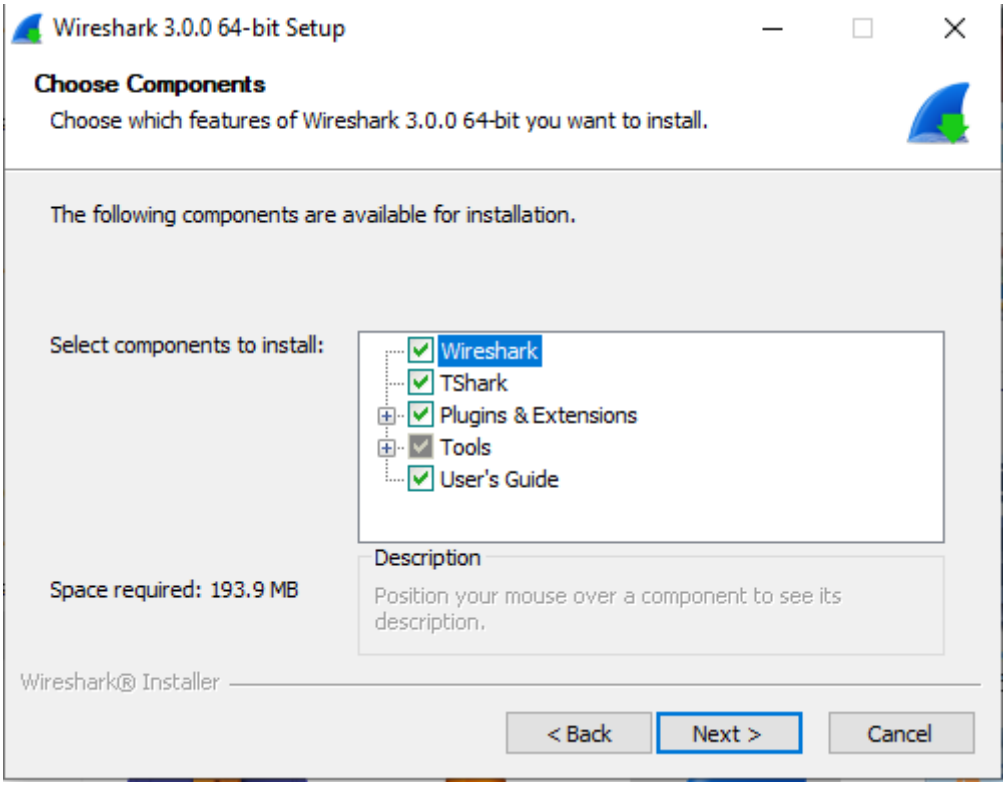
Após escolher qual versão é compatível com o computador onde irá ser instalada, o *download* pode ser feito apenas dando um clique em cima do nome do pacote instalador. Assim, a página a seguir irá ser aberta.



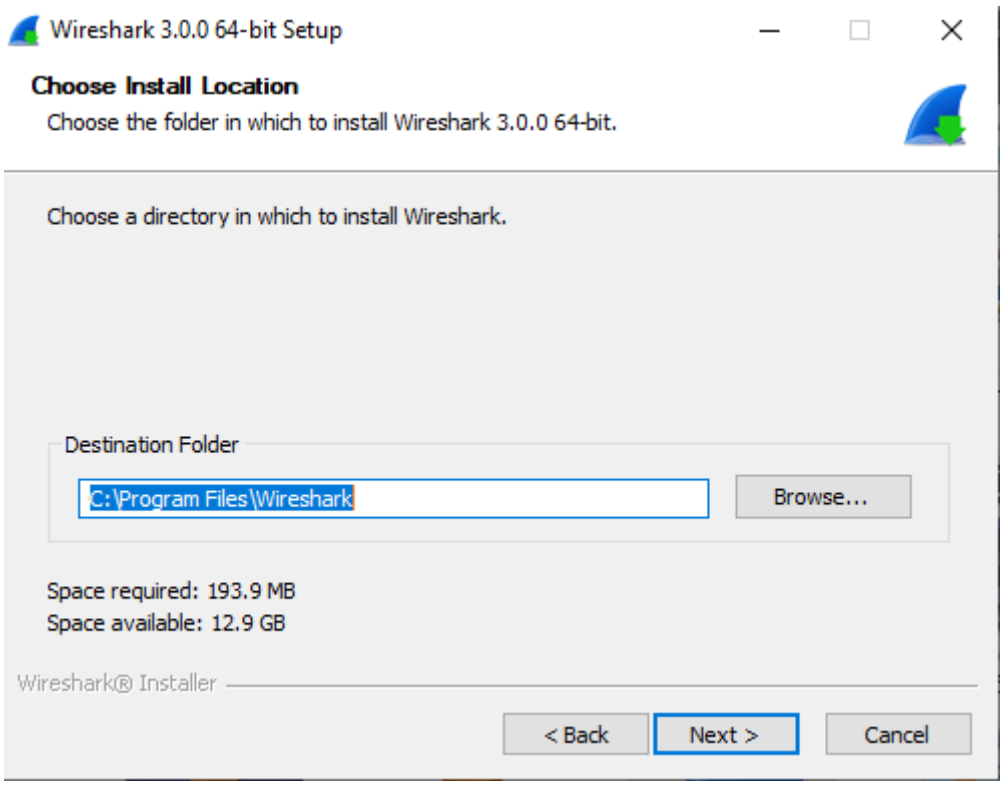
Nesta tela, algumas informações como: versão e configuração (32 ou 64 *bit*) foram escolhidas. Ao clicar no botão *next* (próximo) a instalação irá começar a ser feita, mas antes é preciso concordar com vários termos de uso.



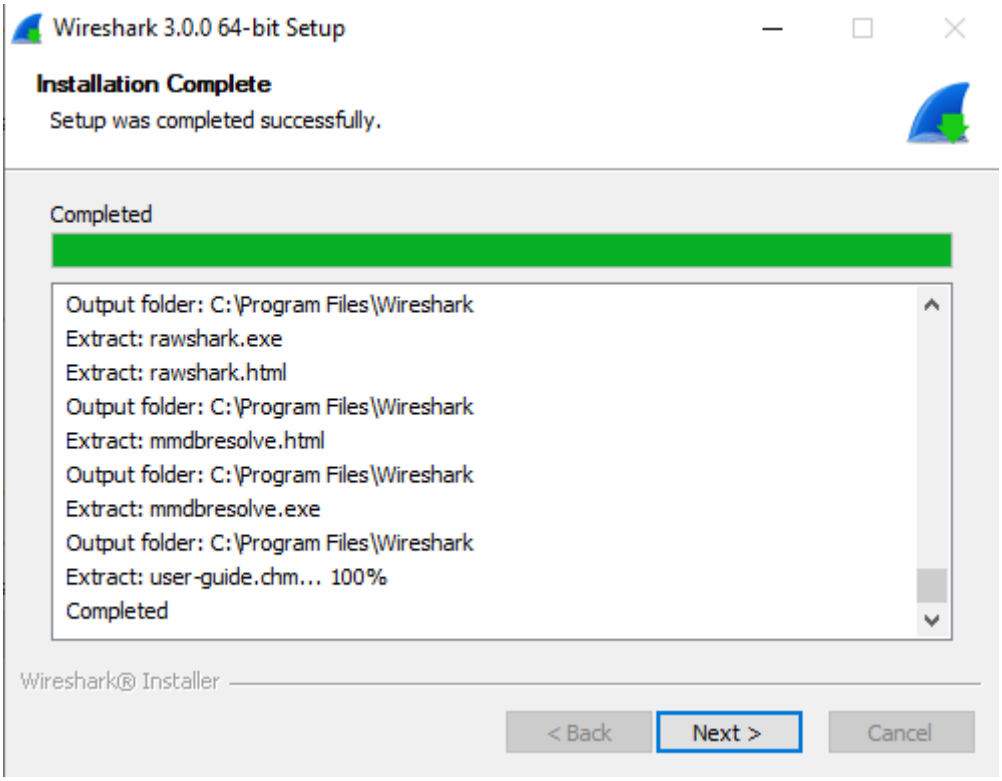
A instalação só poderá ser feita caso o usuário aceite os termos, caso contrário, a instalação será cancelada.



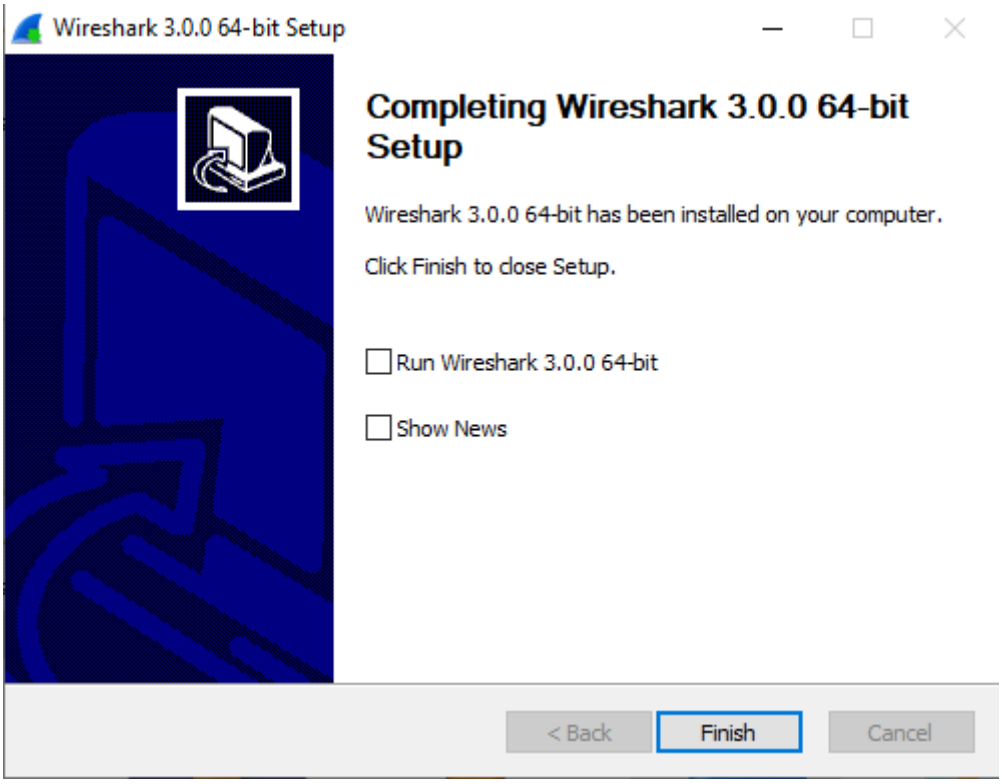
Acima temos os componentes que serão instalados juntos com a ferramenta *wireshark*, visto que, alguns desses componentes são pertinentes para o funcionamento correto da ferramenta.



É preciso também escolher o local onde que será salvo a pasta com os componentes e a ferramenta. Na maioria dos casos, este campo já virá preenchido, mas pode alterar o local de destino. Desta forma, a instalação começará.



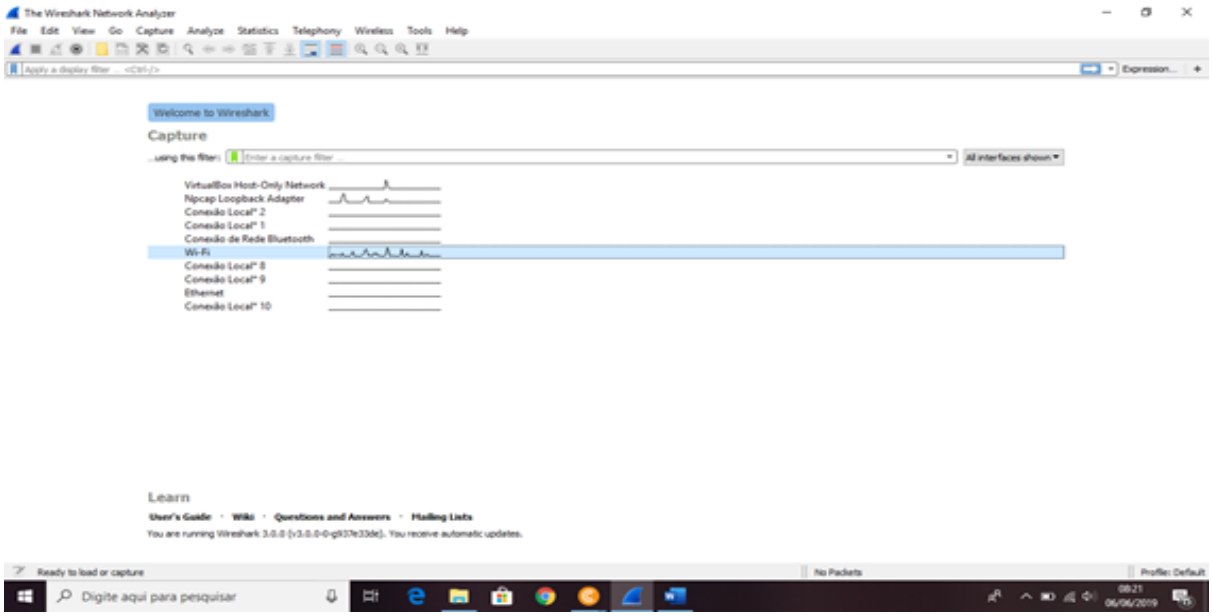
Para concluir a instalação será baixado todos os componentes no local de destino anteriormente escolhidos. Leva alguns minutos para baixar todas as pastas necessárias.



Após *download* completo, pode-se escolher finalizar abrindo ou não a aplicação, ou seja, a ferramenta *wireshark*. Ao iniciar a ferramenta, o primeiro passo é escolher a rede que vai analisar. Lembrando, a versão utilizada neste trabalho é a 3.0.4. A interface pode variar de acordo com a versão.

Figura 2 - Página inicial do *wireshark*

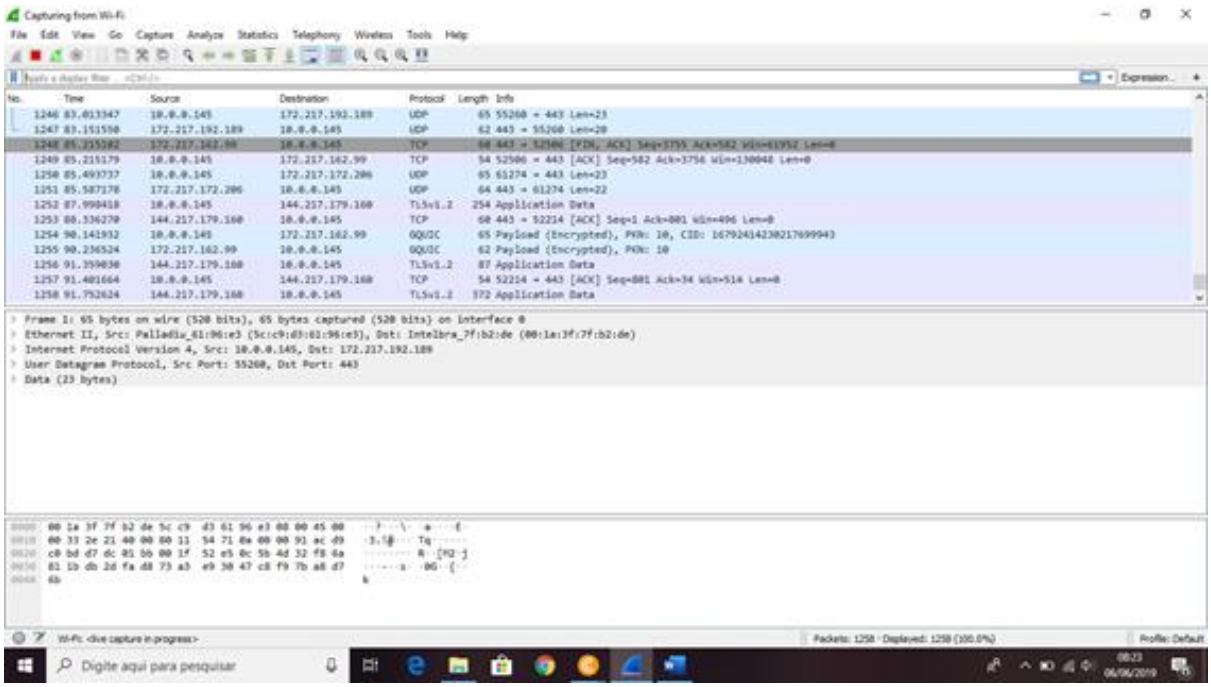
Fonte: Print da tela do wiresh



ark - Elaborada pelo autor

A rede sem fio escolhida foi o *WI-FI*, mas pode-se escolher qualquer uma que esteja disponível e sendo usada, ou seja, a rede em que o computador está conectado. Após escolher, outra página é aberta, revelando os dados que vão sendo capturados em tempo real.

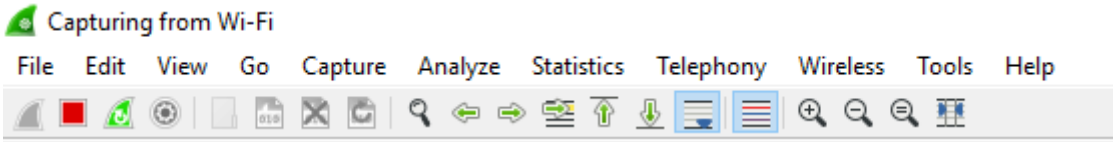
Figura 3 - Captura de pacotes



Fonte: Print da tela do wireshark - Elaborada pelo autor

Em poucos segundos são feitos dezenas de captura de pacotes que estão passando pela rede, mas vai variar de acordo com o tráfego. Assim sendo, pode-se aparecer mais ou menos dados.

Figura 4 - *Menu* principal

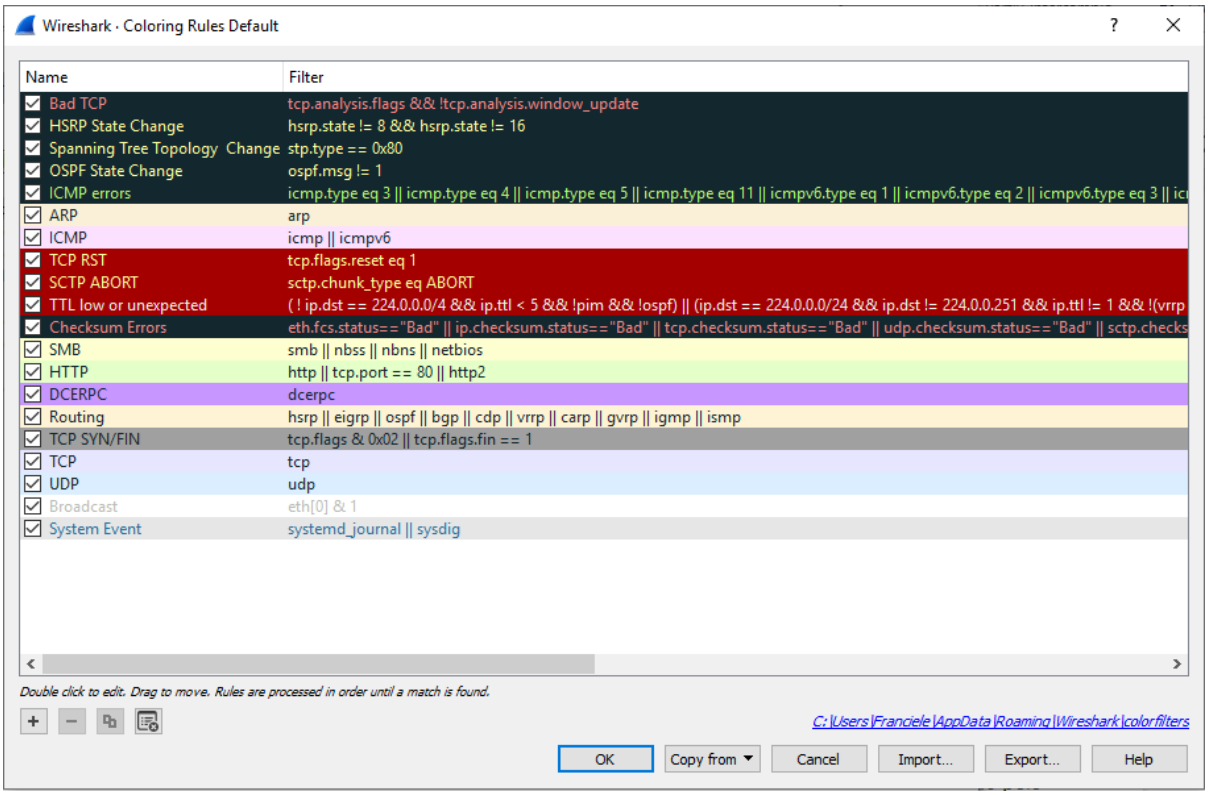


Fonte: Print da tela do wireshark - Elaborada pelo autor

Este é o painel de controle, por onde consegue-se iniciar ou parar uma captura, mudar regras de coloração, salvar os dados já capturados, ver gráficos, entre outras. Não entraremos nas funcionalidades da ferramenta, pois o foco neste trabalho é mostrar pacotes capturados que contém informações confidenciais de terceiros.

O *wireshark* também possui as cores definidas para cada tipo de protocolo, mas o usuário pode alterar essas regras de coloração da forma que achar melhor, adicionando novas regras. Antes de começar qualquer captura, é viável, primeiramente, acessar essas regras de cores. Podem ser vista no *menu*: *View* (visualizar), *Coloring Rules* (regras de coloração). Uma nova página irá abrir, mostrando quais as cores de cada protocolo já definido, podendo ser alterados pelo usuário.

Figura 5 - Regras de coloração do *wireshark*

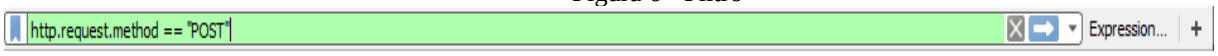


Fonte: Print da tela do wireshark - Elaborada pelo autor

Nesta página, pode-se alterar as regras que já vem criadas - por exemplo, todos os protocolos possuem sua própria coloração, varias cores podem se repetir para diferentes protocolos - da forma que o usuário preferir além de poder criar novas regras. Uma funcionalidade bastante usada por profissionais de rede, que possuem um conhecimento maior sobre as redes de computadores, desta forma, criam novas regras para um melhor monitoramento das redes.

Outra preferência muito útil é utilizar o filtro, para buscar o que for mais indispensável. Auxiliando na pesquisa de algum protocolo ou pacote específico.

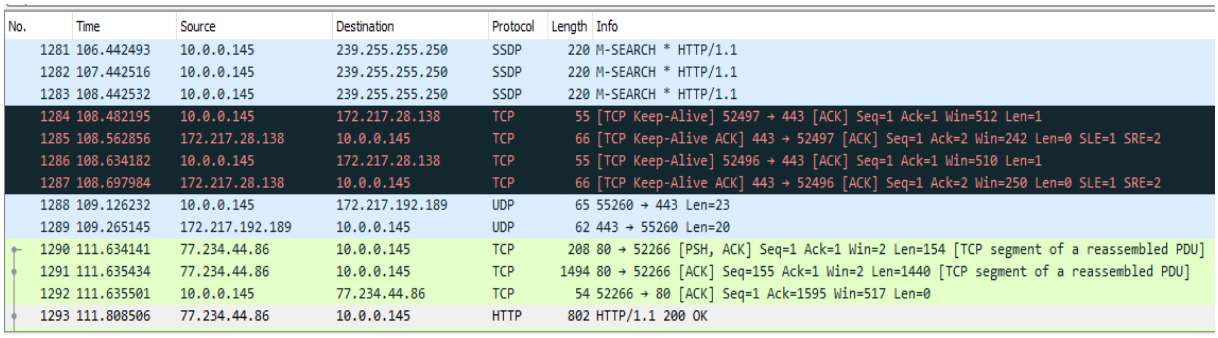
Figura 6 - Filtro



Fonte: Print da tela do wireshark - Elaborada pelo autor

Mais abaixo fica o campo onde mostra-se todos os pacotes capturados. Junto são fornecidas informações como: número do pacote, tempo que demorou para fazer a captura, a origem, o destino, o tipo de protocolo que é, o tamanho e informações adicionais.

Figura 7 - Visualização de pacotes capturados

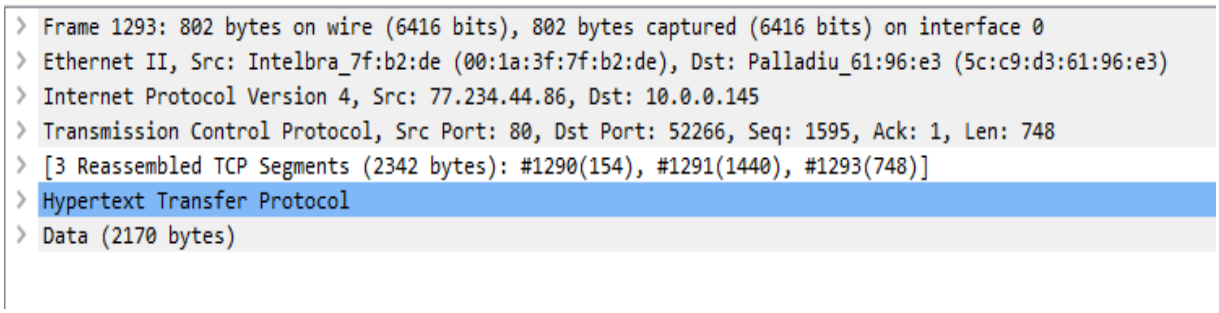


No.	Time	Source	Destination	Protocol	Length	Info
1281	106.442493	10.0.0.145	239.255.255.250	SSDP	220	M-SEARCH * HTTP/1.1
1282	107.442516	10.0.0.145	239.255.255.250	SSDP	220	M-SEARCH * HTTP/1.1
1283	108.442532	10.0.0.145	239.255.255.250	SSDP	220	M-SEARCH * HTTP/1.1
1284	108.482195	10.0.0.145	172.217.28.138	TCP	55	[TCP Keep-Alive] 52497 → 443 [ACK] Seq=1 Ack=1 Win=512 Len=1
1285	108.562856	172.217.28.138	10.0.0.145	TCP	66	[TCP Keep-Alive ACK] 443 → 52497 [ACK] Seq=1 Ack=2 Win=242 Len=0 SLE=1 SRE=2
1286	108.634182	10.0.0.145	172.217.28.138	TCP	55	[TCP Keep-Alive] 52496 → 443 [ACK] Seq=1 Ack=1 Win=510 Len=1
1287	108.697984	172.217.28.138	10.0.0.145	TCP	66	[TCP Keep-Alive ACK] 443 → 52496 [ACK] Seq=1 Ack=2 Win=250 Len=0 SLE=1 SRE=2
1288	109.126232	10.0.0.145	172.217.192.189	UDP	65	55260 → 443 Len=23
1289	109.265145	172.217.192.189	10.0.0.145	UDP	62	443 → 55260 Len=20
1290	111.634141	77.234.44.86	10.0.0.145	TCP	208	80 → 52266 [PSH, ACK] Seq=1 Ack=1 Win=2 Len=154 [TCP segment of a reassembled PDU]
1291	111.635434	77.234.44.86	10.0.0.145	TCP	1494	80 → 52266 [ACK] Seq=155 Ack=1 Win=2 Len=1440 [TCP segment of a reassembled PDU]
1292	111.635501	10.0.0.145	77.234.44.86	TCP	54	52266 → 80 [ACK] Seq=1 Ack=1595 Win=517 Len=0
1293	111.808506	77.234.44.86	10.0.0.145	HTTP	802	HTTP/1.1 200 OK

Fonte: Print da tela do wireshark - Elaborada pelo autor

Ao clicar em um determinado pacotes, mais informações sobre ele são disponibilizadas.

Figura 8 - Campo de informações adicionais



Fonte: Print da tela do wireshark - Elaborada pelo autor

Ao clicar na seta ao lado de cada tópico, várias outras informações podem ser vista. Como mencionado anteriormente, pessoas que não possuem um prévio conhecimento não conseguiram ler e entender o conteúdo.

Como supracitado anteriormente, pacotes contendo informações podem ser capturados através do ataque *sniffing*, possibilitando a leitura de muitas informações. Para poder se proteger deste tipo de ataque, proporcionado pela ferramenta *wireshark*, a utilização de VPN é uma solução, pois, ao se conectar na rede usando uma VPN, os pacotes que forem enviados pela *internet*, mesmo que capturados não terá seus dados lido porque estão criptografados, garantindo o sigilo da informação enquanto está trafegando dá origem ao destino.

2.5 VPN

VPN, do inglês *Virtual Private Network* (Rede Virtual Privada) é uma rede virtual

onde duas redes podem se conectarem, de forma segura, utilizando um canal público de comunicação, criando túneis, por onde transmitem as informações criptografadas entre as redes (GUIMARAES, LINS, DA OLIVEIRA, 2006).

Esses sistemas utilizam criptografia e outros mecanismos de segurança para que assim garanta que somente usuários autorizados possam ter acesso à rede privada e que nenhum dado será lido enquanto estiver passando pela rede pública (CERT.br, 2017).

Segundo Nakamura (2007, p.332), "Com as VPNs, é possível criar conexões privadas, de modo que as comunicações podem passar a ser realizadas por meio de uma única ligação com a rede pública." Ou seja, a VPN garante que os pacotes enviados serão entregues ao destinatário sem que sofram qualquer tipo de dano ou mudança no seu conteúdo mesmo trafegando por uma rede pública, onde ataques de pessoas mal intencionadas são constantes.

Possibilita também a autenticidade e a cifragem do tráfego nos servidores HTTP que são vulneráveis, pois estabelece uma camada extra de segurança, criptografando as pesquisas feitas ou os dados confidenciais digitados nos campos de algum *site* (RUFINO, 2015) .

As VPNs utilizam fundamentos de criptografia e tunelamento, visando garantir a integridade, autenticidade e o sigilo das conexões feitas (NAKAMURA, 2007).

A criptografia é usada para proteger o conteúdo dos pacotes que estão trafegando, ou seja, manter as mensagens seguras. Integridade, sigilo, autenticidade e não-repúdio[27] são propriedades da criptografia para garantir as comunicações, o armazenamento e as transações seguras, que são essenciais hoje em dia (NAKAMURA, 2007).

Segundo Nakamura (2007, p. 301), "A cifragem (*encryption*) é o processo de disfarçar a mensagem original, o texto claro. [...] enquanto a decifragem (*decryption*) é o processo de transformar o texto cifrado de volta em texto claro original". A VPN utiliza a criptografia para ocultar a mensagem que irá ser enviada para o destinatário, sendo este o único que pode descriptografar a mensagem e ver seu conteúdo, pois tem a chave secreta que decodifica os dados da mensagem. Existem dois tipos de chave: privada e pública.

A chave privada ou simétrica, segundo Nakamura (2007, p. 302), "é responsável pelo sigilo das informações por meio da utilização de uma chave secreta para a codificação e decodificação dos dados". Desta forma, se algum pacote for capturada não poderá ser aberto por não ter a chave simétrica certa para fazer a descriptografia.

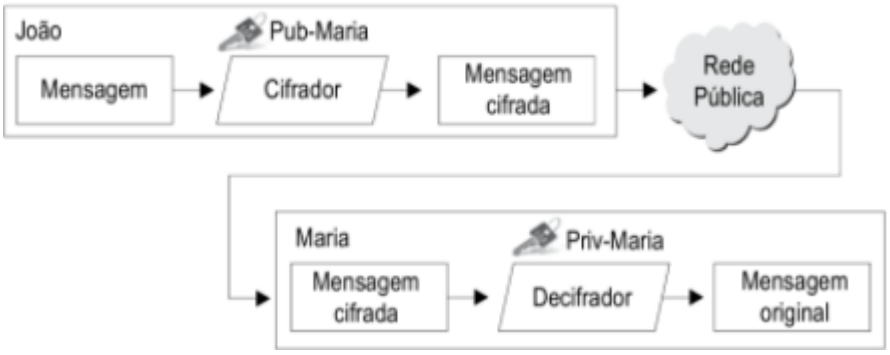
[27] Não-repúdio - não negar a execução de uma ação particular.

Figura 9 - Criptografia com chave privada ou simétrica

Fonte: Nakamura (2007, p. 303)[28]

A chave pública ou assimétrica, possibilita, não somente o sigilo da mensagem, mas também a autenticidade, o não-repúdio e a integridade, visando a comunicação segura. De acordo com Nakamura (2007, p. 303), "Uma mensagem [...] pode ser cifrada utilizando-se uma chave pública e decifrada utilizando-se somente a chave privada correspondente ou vice-versa".

Figura 10 - Criptografia com chave pública ou assimétrica



Fonte: Nakamura (2007, p. 304)

A criptografia prevê o sigilo da comunicação, uma vez que, as chaves são geradas aleatoriamente, impossibilitando adivinhas as chaves futuras por causa de familiaridades, e precisam ser trocadas com frequência, para que assim, ataques de *crackers* tenham menores chances de darem resultados. Assim, garantindo uma maior segurança na navegação pela *internet.*

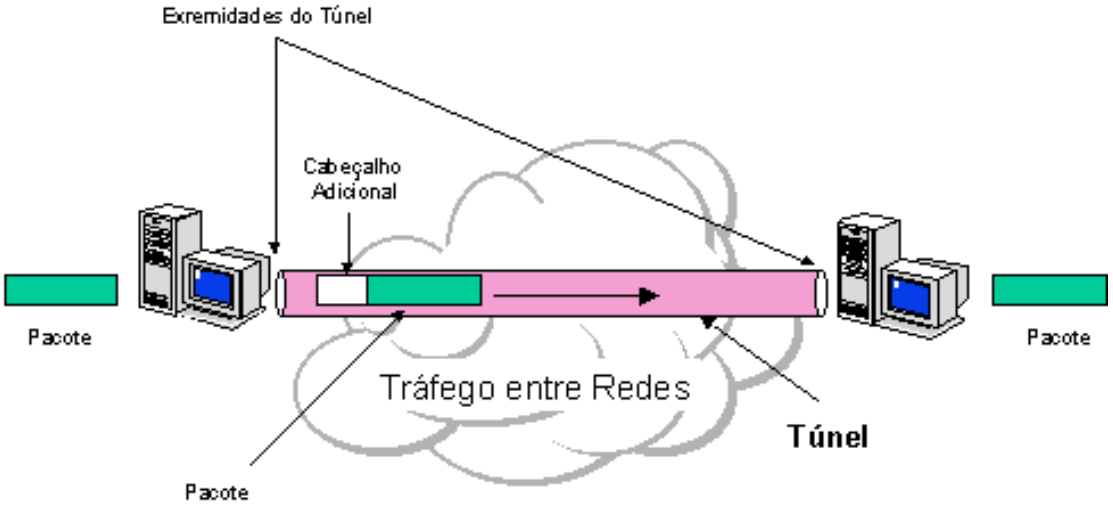
Outro fator importante da VPN é o tunelamento, que possibilita a comunicação entre organizações, ou seja, a troca de informações será realizada, enviada e recebida, independente do tipo de protocolo por onde a informação esteja trafegando em redes diferentes

[28] NAKAMURA, Emilio Tissato; DE GEUS, Paulo Lício. **Segurança de redes em ambientes cooperativos**. Novatec Editora, 2007.

(NAKAMURA, 2007).

Ao enviar alguma mensagem, primeiro se criptografa e depois encapsula e coloca o original dentro de um novo pacote. Essa mensagem irá passar dentro do tunelamento. Daí a ideia de túnel, pois o pacote sai de uma extremidade até a outra da conexão. Ilustrado na imagem abaixo (GUIMARAES, LINS, DA OLIVEIRA, 2006).

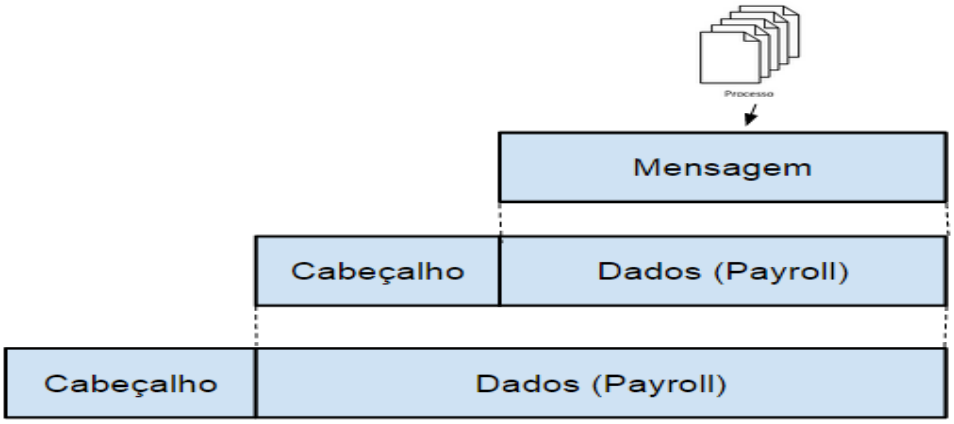
Figura 11 - Tunelamento



Fonte: Redes VPN e IPSec[29]

O cabeçalho adicional contém as informações de transporte, como: a origem de onde saiu e o destino, para onde vai. Ao ser encapsulado, as informações iniciais que serão enviadas são colocadas dentro de outro pacote, que também terá outras informações de transporte. Estas, por suas vez, são do destino onde o tunelamento da VPN termina. Ao chegar no destinatário, o desencapsulamento é feito, ou seja, as informações são retiradas de dentro do pacote e direcionada para o usuário final (FOROUZAN, 2009).

Figura 12 - Encapsulamento



[29] Redes VPN e IPSec - https://www.gta.ufrj.br/grad/04_1/vpn/Figuras/RDITunalamento01.gif

Fonte: Bóson Treinamentos[30]

O encapsulamento insere a mensagem dentro de um outro pacote com cabeçalho - informações de transporte. Essas informações podem ser encapsuladas diversas vezes e a cada novo encapsulamento, novas informações de transportes são inseridas no cabeçalho ou rodapé do pacote. A mensagem inicial será transmitida sem sofrer alterações apesar de receber informações adicionais. Quando chega ao destinatário, de acordo com os dados de transporte de destino, o pacote será desencapsulado, tirando as novas informações de cabeçalho e rodapé adicionados anteriormente. Por fim, sobrando apenas a mensagem que foi inicialmente enviada (SANDERS, 2010).

Ao utilizar uma VPN é possível sim, mesmo estando navegando por uma rede vulnerável ou monitorada, ocultar os dados e informações, mediante criptografia. Desta forma, *crackers* não conseguiram obter dados confidenciais de terceiros.

[30]http://www.bosontreinamentos.com.br/redes-computadores/curso-de-redes-camada-de-transporte-da-pilha-tcpip/

3 METODOLOGIA

A metodologia trata do estudo de métodos e procedimentos que são realizados para chegar a um determinado fim. Assim sendo, são as técnicas que serão seguidos para a realização do presente trabalho, para que assim, alcance os objetivos elencados para o desenvolvimento do mesmo.

3.1 Elaboração do trabalho

O tipo de pesquisa utilizada no presente trabalho é a pesquisa bibliográfica que, segundo VERGARA (1998), "é o estudo sistematizado desenvolvido com base em materiais publicados em livros, revistas, jornais, redes eletrônicos [...]". Mediante esta investigação coletará informações, dados e conceitos pertinentes e de embasamento para a construção e desenvolvimento da solução para a problemática levantada neste projeto de pesquisa.

Outro tipo de pesquisa utilizada, será a descritiva, pois, segundo VERGARA (1998) expõe características. Neste trabalho, características sobre o *software wireshark*, que captura pacotes variados, que pode conter dados confidenciais, que passam em texto claro pela rede; e a ferramenta VPN, que pode ser usada para proteger essas informações que trafegam por uma rede sem fio de computador.

Além de possuir uma investigação exploratória. Ainda segundo VERGARA (1998), "A investigação exploratória é realizada em área na qual há pouco conhecimento acumulado e sistematizado". Para se obter uma proximidade com o objeto do estudo, é preciso fazer esta investigação, pois, é mediante ela que chegará a problemática e as possíveis soluções. Ao explorar o objeto, levantando dados através da pesquisa bibliográfica, o conhecimento irá aumentar.

Segundo O CERT.br (2019), "O que define as chances de um ataque na Internet ser ou não bem sucedido é o conjunto de medidas preventivas tomadas pelos usuários, desenvolvedores [...] e administradores dos computadores [...]".

Diante disto, a pesquisa de laboratório, que segundo VERGARA (1998), "é experiência realizada em local circunscrito [...] Simulações em computador situam-se nesta classificação". Será realizada simulação em ambiente computacional, para registrar, analisar e interpretar os dados capturados do protocolo HTTP vulnerável pela ferramenta *wireshark*.

O protocolo HTTP é usado em toda a Word Wide Web (www). Segundo Tanenbaum (1997, p.493), "Ele especifica as mensagens que os clientes podem enviar aos servidores e que respostas eles receberão". Uma vez que, o HTTP especifica a estrutura dessa mensagens e o modo como elas devem ser trocada.

O HTTP usa o TCP como protocolo de transporte. Desta forma, o TCP oferece ao HTTP um serviço confiável de transferência de dados, o que implica que toda mensagem de requisição HTTP emitida por um processo cliente chegará intacta ao servidor, e vice-versa, pois toda mensagem de resposta HTTP emitida pelo processo servidor chegará intacta ao cliente (KUROSE; ROSS, 2006).

Será realizado dois testes para uma melhor demonstração da problemática e da solução propostas pelo presente trabalho. Os testes ocorreram da seguinte forma:

3.1.1 Teste 1 - VirtualBox

VirtualBox é um produto gratuito, de código aberto, onde é possível criar uma máquina virtual com qualquer sistema operacional desejado, possibilitando que vários sistemas operacionais sejam usados em um mesmo computador.

Com o VirtualBox instalada em uma computador físico, a criação de uma máquina virtual é necessária, pois nela a ferramenta *wireshark* irá ficar capturando tudo o que se passa pela rede onde está conectada junto com o computador físico. Desta forma, tudo que o computador físico acessar, a máquina virtual saberá. Pesquisas em *sites* HTTP vulneráveis serão feitas para demonstrar se é possível obter informações através da captura de pacotes.

Após feito, o teste será refeito. Desta vez, o computador físico estará utilizando a ferramenta VPN, para criptografar as informações que trafegam dentro dos pacotes.

3.1.2 Teste 2 - Computador Físico

O segundo teste consistirá em: um computador físico, com a ferramenta *wireshark* instalada e capturando, irá disponibilizar *internet* através do *hotspot*[31]. Após a distribuição da rede sem fio, vários *smartphones*[32] se conectaram nesta rede para poderem fazer buscas pelos navegadores em *sites* HTTP.

A ferramenta *wireshark* irá capturar esses pacotes para que assim, possa ser feita a análise a procura das informações que foram trocadas.

O mesmo teste será refeito, mas com os *smartphones* utilizando alguma VPN, para mostrar se o que é proposto pelas VPNs são mesmo cumpridas, ou seja, a ocultação dos dados que serão trocados entre dispositivos pela rede.

---

[31] Hotspot - um ponto de acesso que serve para se ligar a rede, neste caso, pelo *WI-FI*.

[32] Smartphones - celulares inteligentes.

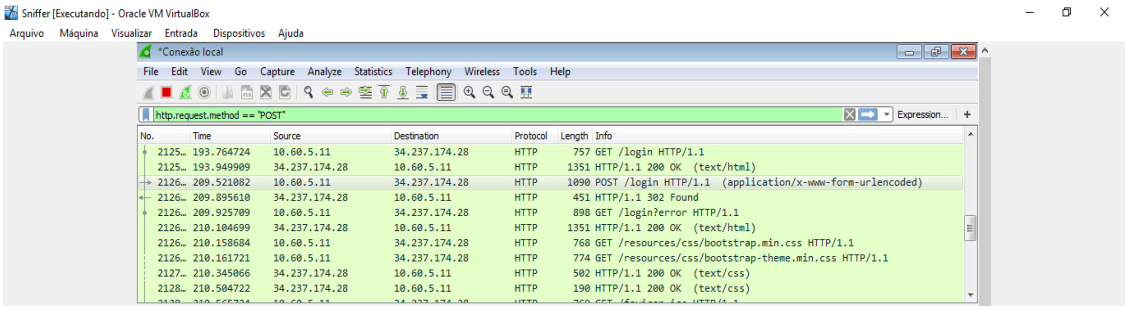
4 RESULTADOS E DISCUSSÕES

A partir das informações relevantes que foram obtidas através da especificação de conceitos variados a elaboração do teste de laboratório em um ambiente computacional será possível, pois é um campo controlado, onde os testes realizados foram precisos para a resolução da problemática e o alcance dos objetivos.

4.1 Teste 1 - VirtualBox

O primeiro teste, como mencionado anteriormente, envolve o uso de uma máquina virtual com a ferramenta *wireshark* executando e um computador físico, onde as pesquisas serão feitas em *sites* vulneráveis com protocolo HTTP para transferir dados na *internet*. Como a máquina virtual está conectada na mesma rede que o computador físico, a captura dos pacotes que trafegam é possível.

Figura 12 - Pacotes Capturados



Fonte: Print da tela do wireshark - Elaborada pelo autor

Todos os pacotes que estão trafegando pela rede que o computador físico está conectado estão sendo capturados pela ferramenta. Como o foco do presente trabalho são os HTTP vulneráveis que ainda existem na internet, a utilização de filtro é essencial. O protocolo HTTP não possui a camada adicional de segurança SSL/TLS como o HTTPS. Deste modo, as aplicações que acontecem entre cliente e servidor, na maioria das vezes, não são criptografados, passando em texto claro pela rede. Sendo assim, *crackers* podem ler essas informações.

Para se obter pacotes com informações de usuários ou senha, o filtro mais adequado envolve o método POST, mas para que isso seja possível, primeiramente, o computador se conecta em um determinado *site*, depois de alguns segundos recebe-se um retorno para sua solicitação por meio do método GET.

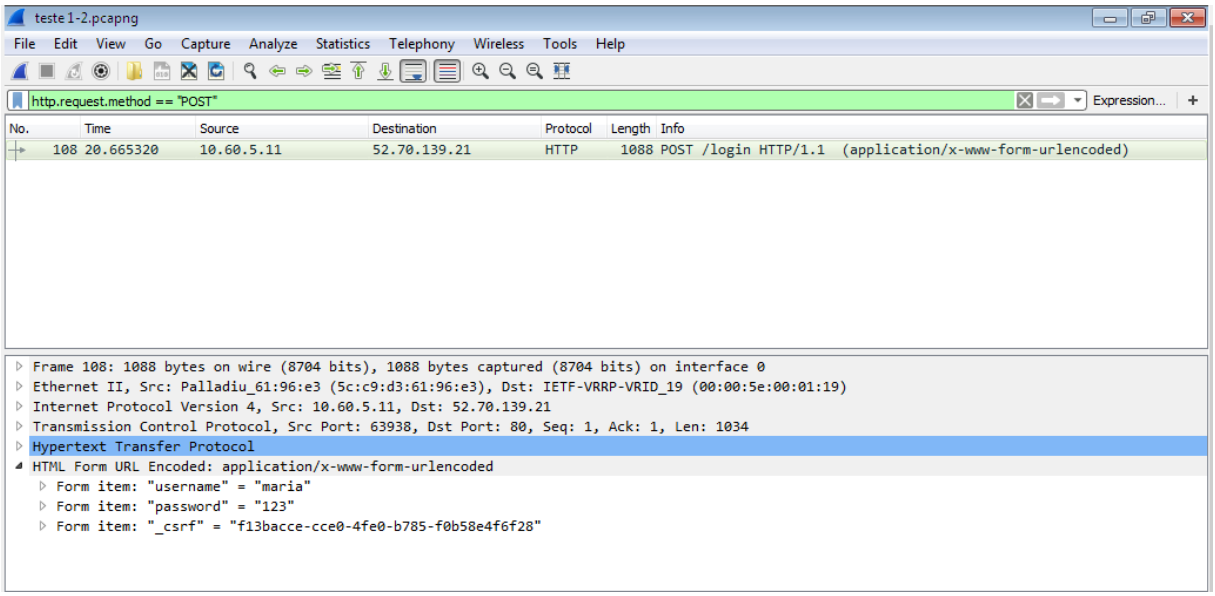
O método GET é usado para recuperar dados ou executar uma consulta em um recurso, algum *site* ou plataforma *on-line*. Os dados retornados do *webservice* é uma representação do recurso solicitado, ou seja, obter informações de algum servidor. Já o

método POST é usado para criar um novo recurso. Um exemplo: preencher e enviar um formulário do *site*. O serviço na *web* pode responder com os dados ou estados que indicam o sucesso ou fracasso do envio do formulário. (DE PAIVA FERREIRA; MATIUSSO JR, 2010, p.5)

Para se obter apenas o protocolo HTTP com resposta do método POST, utiliza-se o filtro: **http.request.method == "POST".** Só vão aparecer os pacotes que foram enviados através de métodos de solicitação que o HTTP recebeu por meio do POST.

Após simplificar os resultados, fica mais fácil analisar os pacotes restantes. Ao selecionar um, podemos ver mais informações no campo abaixo:

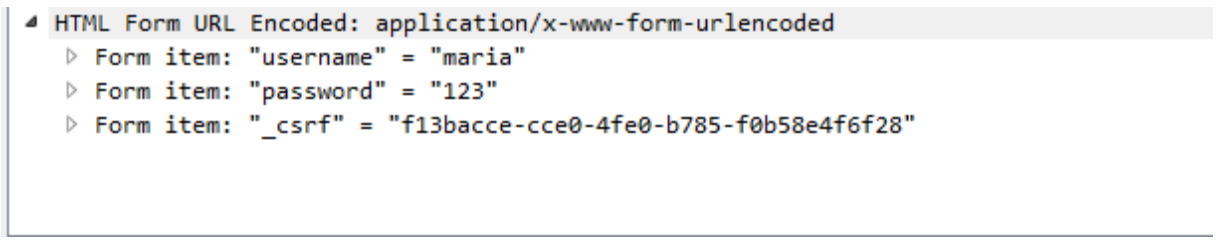
Figura 13 - Filtro http



Fonte: Print da tela do wireshark - Elaborada pelo autor

No último ícone, HTML *Form* URL *Encoded* (HTML do URL codificado), informações de campos que foram preenchidos no *site*, como *username* (usuário) e *password* (senha), podem ser visualizados. É desta forma que pessoas mal intencionadas obtém informações de terceiros, podendo usá-las para fazerem outras invasões, chegando até descobrirem dados bancários.

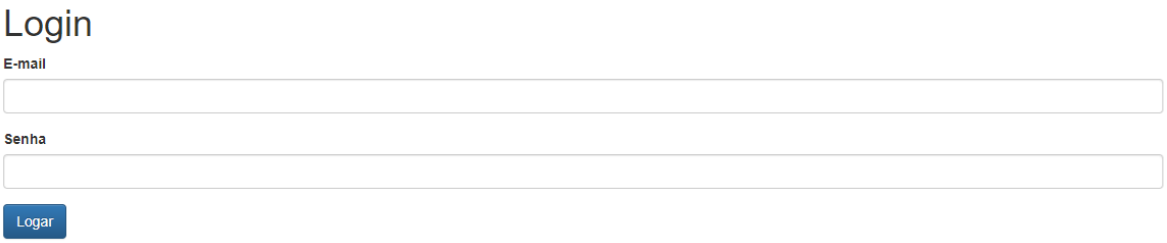
Figura 14 - Ícone html



Fonte: Print da tela do wireshark - Elaborada pelo autor

O teste acima foi realizado na plataforma disponível em: http://exemplo-login-fraco-alura.herokuapp.com/login. Utilizada apenas para testes que pretendem mostrar a vulnerabilidade que o protocolo HTTP representa na rede.

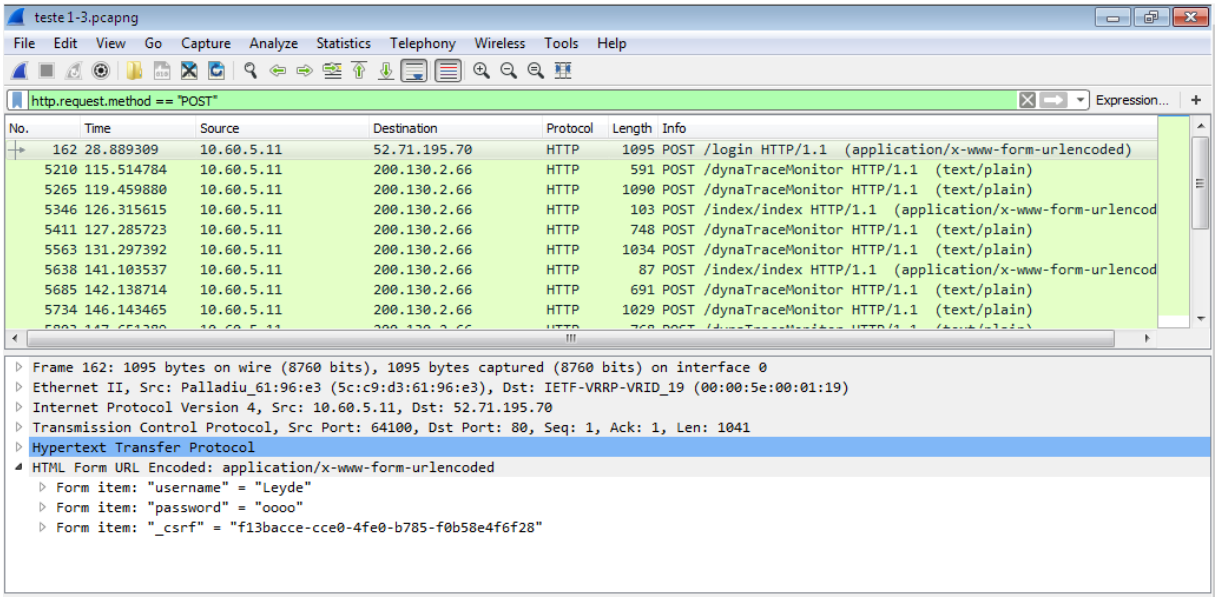
Figura 15 - Teste de login



Fonte: Print da tela do wireshark - Elaborada pelo autor

Esta plataforma pega os dados preenchidos nos campos de *e-mail* e senha, os envia para um servidor onde essas informações serão armazenadas. Nenhuma informação irá retornar ao preencher os campos. Outros *sites* com HTTP também podem ter seus pacotes interceptados e lidos. Assim sendo, revelando dados sigilosos dos usuários que navegam pela *internet*.

Figura 16 - Pacotes capturados com método post



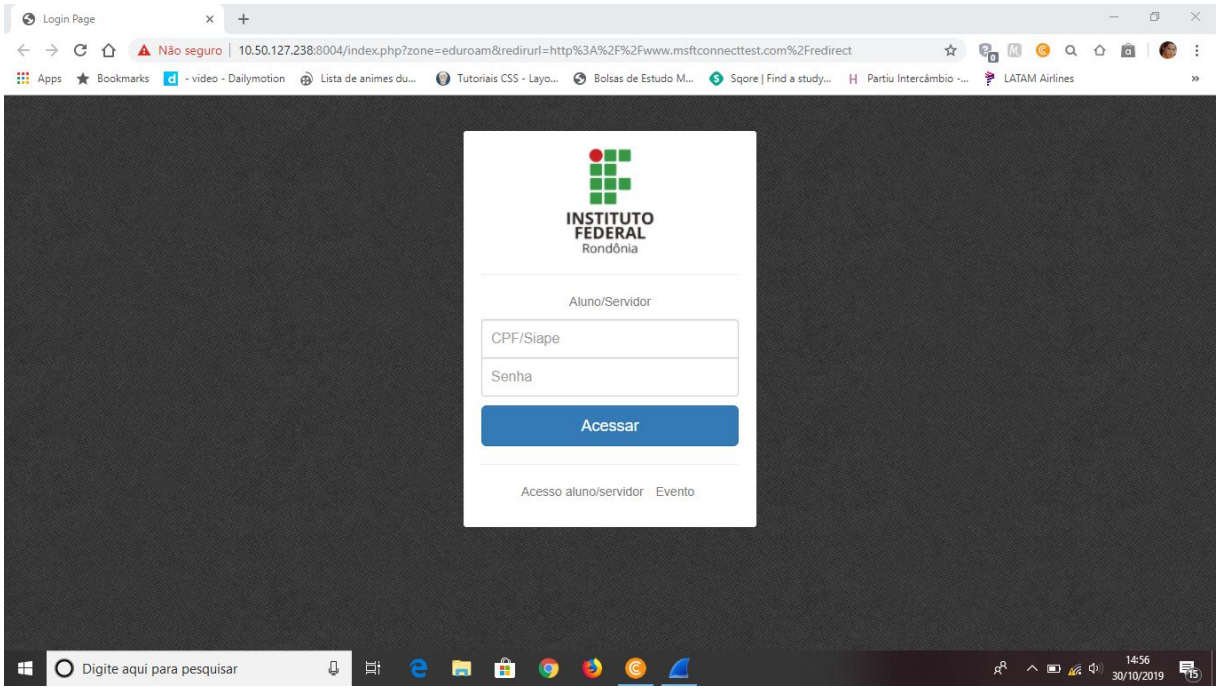
Fonte: Print da tela do wireshark - Elaborada pelo autor

O mesmo teste foi realizado outras vezes, para que assim, pacotes capturados com informações variadas possam provar que o HTTP vulnerável com resposta do método POST, foram lidos, pois as informações estão trafegando em texto claro pela rede.

Uma outra plataforma a ser testada foi a autenticação do serviço do IFRO do *campus* de Ji-Paraná. Para se conectar na rede do IFRO é necessário preencher os dados de identificação: CPF e senha. Este serviço não é seguro pois ocorre mediante protocolo HTTP,

ou seja, não criptografa os dados quando os envia para o servidor para a autenticação, verificação se os dados preenchidos pertencem a alguém que foi previamente cadastrado no servidor do IFRO, assim sendo, um aluno ou servidor. Se a verificação for bem sucedido, o acesso a *internet* local é liberado, senão, o acesso será negado.

Figura 17 - Login do IFRO



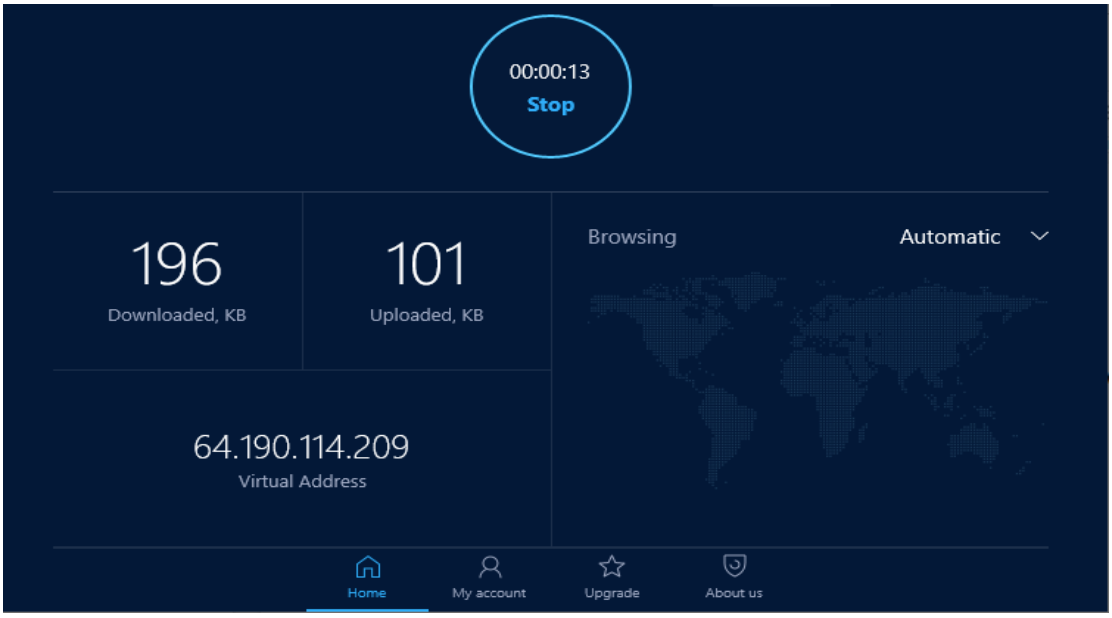
Fonte: Print da tela do wireshark - Elaborada pelo autor

Para prevenir que o *wireshark* não capture os dados confidenciais de terceiros que estão trafegando pela rede no protocolo HTTP, foi escolhido a utilização de uma rede virtual privada - VPN -, uma vez que, a ferramenta VPN garante transmitir pacotes pela rede de forma segura, criptografando os dados.

Após realização de vários testes com diversas VPNs gratuitas, e obtendo o mesmo resultado de omissão em todos, a VPN *Hotspot Shield* foi escolhida para representação do uso. A seguir segue a tela inicial da VPN, que já está executando, ou seja, está conectado em um servidor, fornecendo assim, um novo número de IP virtual para omitir o verdadeiro número do computador que está utilizando a VPN.

A conexão ao provedor foi feita de forma automática, mas é possível escolher o provedor de um determinado país, da preferência do usuário, e a conexão pode ser parada quando preferir.

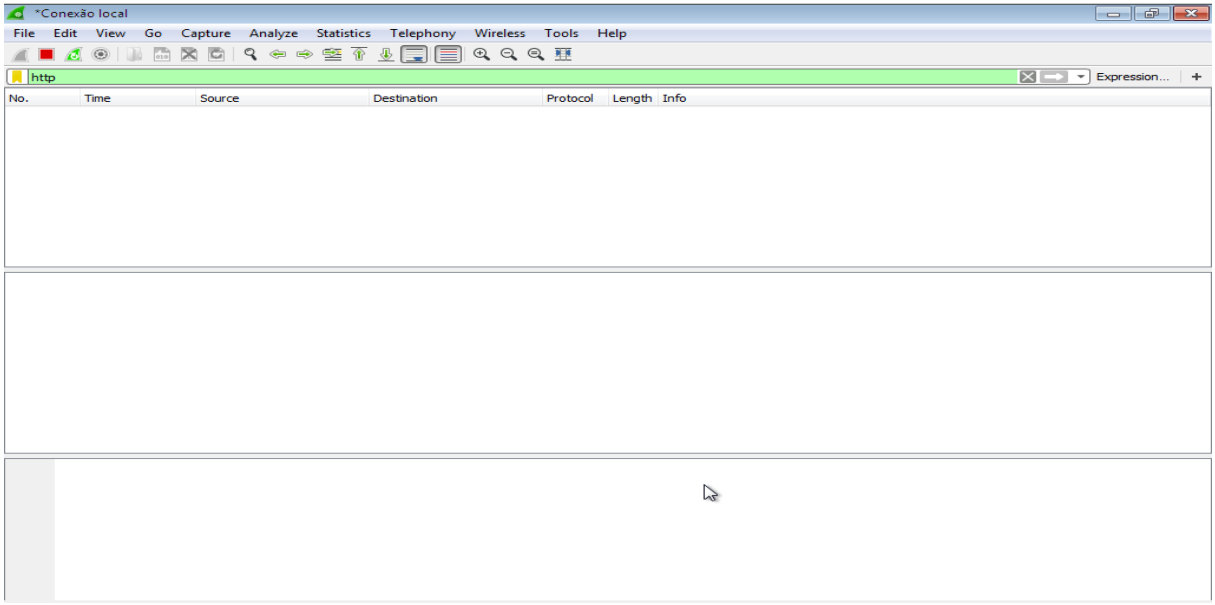
Figura 18 - Tela inicial da VPN



Fonte: Print da tela da VPN - Elaborada pelo autor

Como o propósito do trabalho é a utilização de VPN para ocultar que dados passam as claras pela rede, a figura a seguir mostra que, ao iniciar a VPN, nenhum dado que esteja trafegando na rede no protocolo HTTP vai ser capturado. Uma vez que, a criptografia e o tunelamento permitem que as informações saiam da origem e cheguem ao destino, sem ser capturada por alguma ferramenta de *sniffer*.

Figura 19 - Uso da VPN



Fonte: Print da tela do wireshark - Elaborada pelo autor

Os outros pacotes, que não sejam do protocolo HTTP, continuam sendo capturados normalmente, mas ao usar o filtro HTTP, percebe-se que nenhum pacote foi encontrado, mesmo que, no computador físico testes na plataforma http://exemplo-login-fraco-

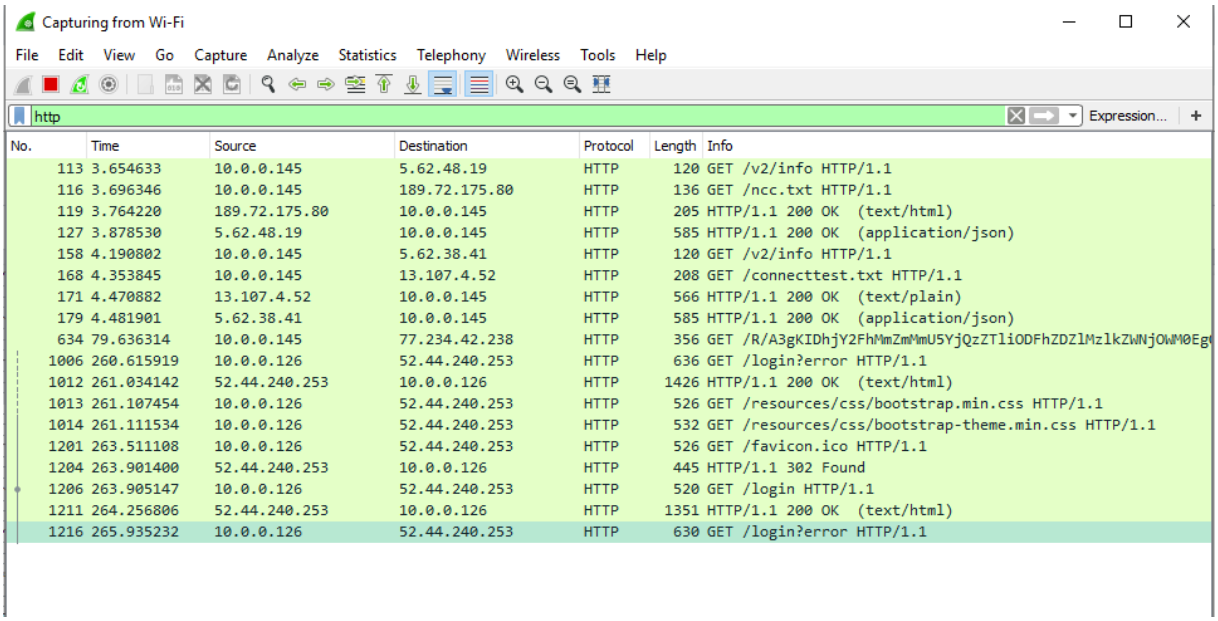
alura.herokuapp.com/login estão sendo feitos ao mesmo tempo que o *wireshark* conectado na rede está capturando. Sendo assim, os pacotes que estão passando pela rede não podem ser capturados e lidos, assim, não revelam dados confidenciais.

4.2 Teste 2 - Computador Físico

No segundo teste, o uso de um computador físico e de vários *smartphone* serão necessários para a realização do mesmo. A ferramenta *wireshark* está instalada e executando no computador físico, e por meio do roteamento da rede por meio do *hotspot* do computador, vários *smartphone* foram conectados ao *WI-FI*, assim, usam a mesma rede.

A plataforma para teste é a mesmo que a anterior, usado em varios celulares diferentes preenchendo os campos de *e-mail* e senha com informações qualquer.

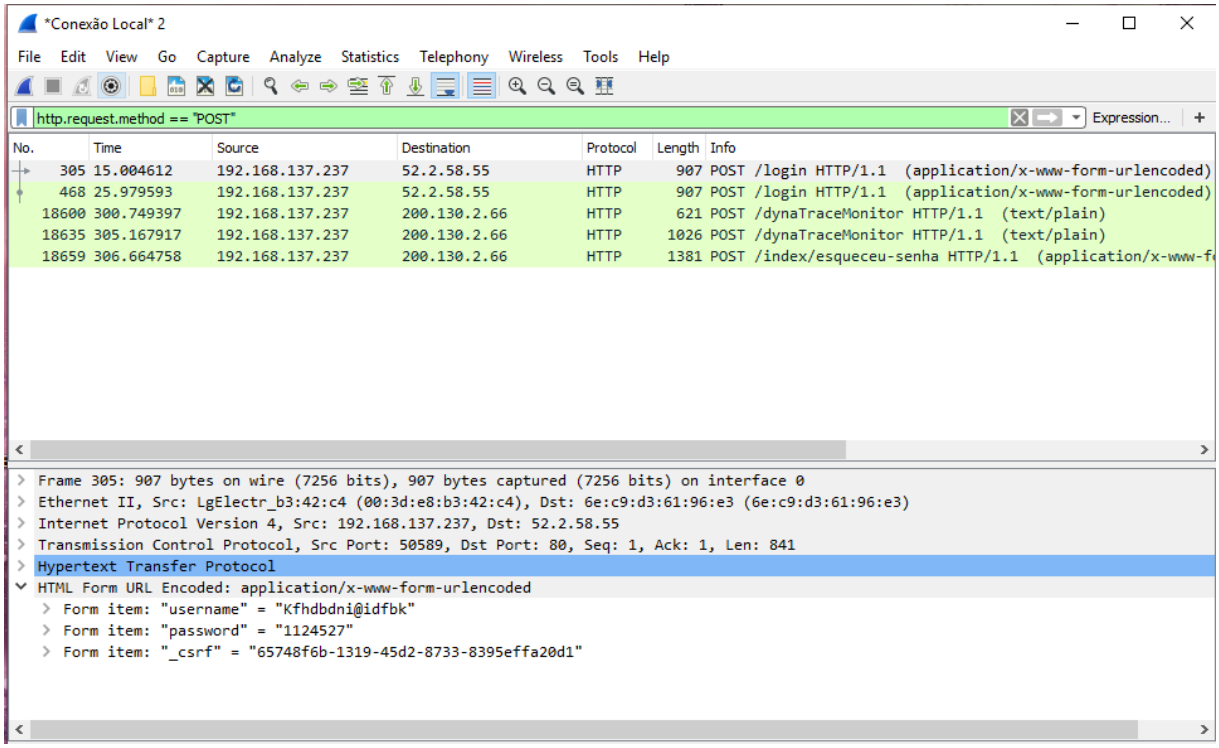
Figura 20 - Pacotes capturados com filtro http



Fonte: Print da tela do wireshark - Elaborada pelo autor

Como percebe-se, vários pacotes foram localizados mesmo usando o filtro HTTP para minimizar os resultados. Ao usar o filtro mais complexo do HTTP, os resultados diminuem perceptivelmente. E, assim como no teste anterior, os dados dos campos que foram preenchidos podem ser visualizados no ícone HTML.

Figura 21 - Filtro http.request.method == “POST”



Fonte: Print da tela do wireshark - Elaborada pelo autor

Pessoas mal intencionadas, que podem estar conectados à mesma rede que um usuário desinformado da vulnerabilidade, que preencha algo em um *site* vulnerável pode ter seus dados lidos de forma antiética, podendo acarretar perdas maiores do que apenas dados de *login.*

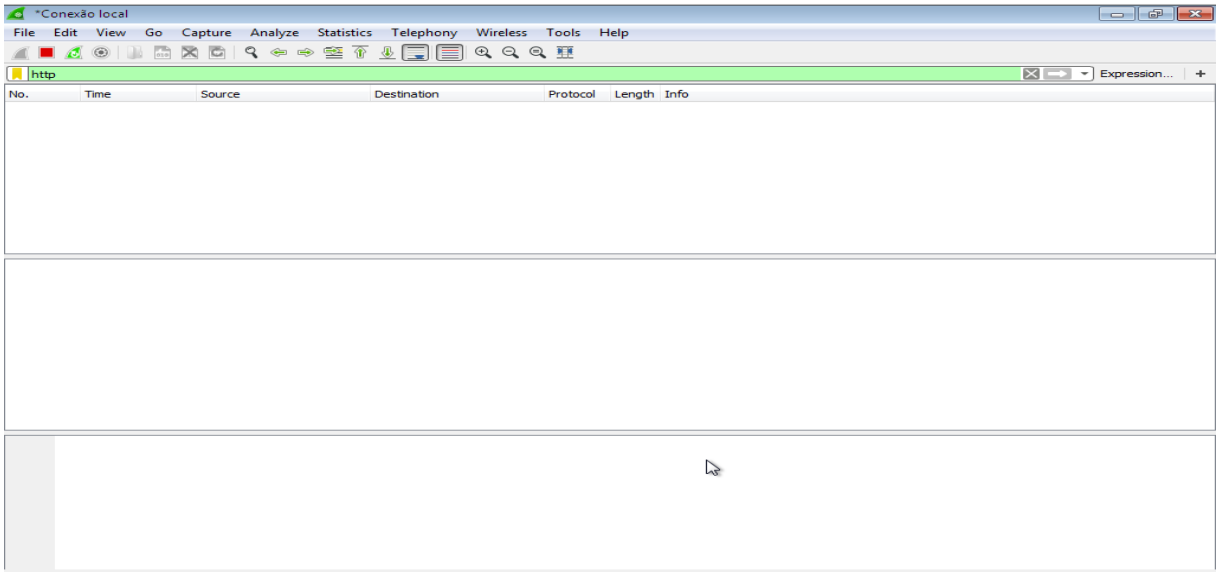
A VPN usada nos dois testes, para demonstração, são a mesma, pois possuem versões *desktop* e *mobile*, além de serem simples de manusear, bons comentários de outros usuários e uma boa avaliação de desempenho e ter a opção de teste gratuito.

Figura 22 - Tela inicial da VPN *mobile*

Fonte: Print da tela da VPN - Elaborada pelo autor

Após se conectar utilizando a VPN, a tela inicial de captura do *wireshark* mostra os pacotes capturados, como no teste anterior, mas ao usar o filtro **http**, a tela fica vazia, como se não estivesse trafegando nenhuma pacote pela rede.

Figura 23 - Uso da VPN 2



Fonte: Print da tela do wireshark - Elaborada pelo autor

Por tudo isso, a confidência nos *sites* com protocolo HTTP está se tornando um embaraço relevante, comprovado por meio da simulação feita anteriormente, pois a transferência de dados que ocorre entre cliente/servidor não está sendo feita de forma segura, porque não criptografa os dados para enviá-los. Assim, o teste permitiu o registro, análise e interpretação de dados capturados, chegando a uma conclusão: pessoas mal intencionadas podem usar a ferramenta para obter informações sigilosas, aproveitando-se da vulnerabilidade que o protocolo HTTP possui ao transportar dados pela rede.

5 CONCLUSÃO

A partir da problemática levantada no início do presente trabalho, usuários podem ser muitas informações roubadas por pessoas mal intencionadas, pois, ao se conectar a uma rede, precisasse ceder informações que serviram de identificação. Essas e outras informações podem ser capturadas por meio da ferramenta *wireshark*, uma vez que, diversos pacotes passam pela rede transportando informações em texto claro.

Para que os pacotes que transportam informações sigilosas passem em texto claro pela rede a ferramenta VPN é uma solução, uma vez que, garante sigilo nas transações, pois criptografa os dados, colocando-os dentro de outro pacote e os enviando por um túnel até o destinatário. Mesmo que este pacote for capturado, seus dados não serão lidos.

Como estar conectado é uma necessidade de mais ou menos 74,9% da população brasileira - em 2017 - (dados do IBGE, 2018), o uso de ferramentas que proporcionam segurança estão sendo cada vez mais buscados por usuários para que os ataques na *internet* não sejam bem sucedidos, já que *crackers* buscam atacar cada vez mais os pontos que ainda são frágeis no mundo digital.

O uso da máquina virtual proporcionam uma comodidade a mais, pois pode ficar executando com uma ferramenta *sniffer* capturando os pacotes que estão trafegando na rede para análise posterior. Visto que, poucas suspeitas são levantadas quanto ao uso de VirtualBox, que se tornaram comum no meio computacional.

Assim sendo, a partir da pergunta inicial: É possível utilizar, em uma rede vulnerável e monitorada, alguma ferramenta que possibilita ocultar os dados trafegados por essa rede? A resposta é sim. E a ferramenta é a VPN, pois ela permite que informações sigilosas não passem em texto claro pela *internet*, deixando a troca de dados mais segura e criptografada.

REFERÊNCIAS

Agência IBGE notícias. **PNAD Contínua TIC 2017: Internet chega a três em cada quatro domicílios do país.** Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/23445-pnad-continua-tic-2017-internet-chega-a-tres-em-cada-quatro-domicilios-do-pais> Acessado em 14 set 2019.

ARNBAK, Axel et al. Colapso de segurança no mercado HTTPS. **Com. do ACM** , v. 57, n. 10, p. 47-55, 2014.

Blog oficial do wireshark. **Sniff Free or Die.** Disponível em: <https://blog.wireshark.org/> Acesso em 10 maio 2019.

BORGES, Fábio; FAGUNDES, Bruno Alves; DA CUNHA, Gerson Nunes. VPN: Protocolos e Segurança. **S/D**, v. 10, 2014.

CERT. **Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil.** Disponível em: <https://www.cert.br/stats/incidentes/> Acesso em 20 maio de 2019.

CANTÚ, Evandro. **Redes de computadores e Internet.** São José, CEFET/SC, 2003.

DA SILVA, Luiz Carlos. Redes Wi-Fi: Estudo do Furto de Sinal. 2010.

DE PAIVA FERREIRA, Esequiel; MATIUSSO JR, Mario. Aplicações RESTful, 2010, p.5.

FARRUCA, Nuno Miguel Galego. **Wireshark para sistemas distribuídos**. 2009. Tese de Doutorado. FCT-UNL.

FIELDING, Roy et al. **Protocolo de transferência de hipertexto - HTTP / 1.1** . 1999.

FOROUZAN, Behrouz A. **Comunicação de dados e redes de computadores**. AMGH Editora, 2009.

GIL, Antônio Carlos, 1946 - **Como elaborar projetos de pesquisa**/Antônio Carlos Gil. - 4. ed. - São Paulo : Atlas, 2002.

GUIMARAES, Alexandre Guedes; LINS, Rafael Dueire; DA OLIVEIRA, Raimundo Correa. **Segurança em Redes Privadas Virtuais-VPNs**. Brasport, 2006.

GOMES, Angelo Ferreira. **Um estudo sobre redes de computadores.** Acesso em 23 maio 2019.

HORA, Evandro Curvelo. **Sobre a Detecção Remota de Sniffers para Detectores de Intrusão em Redes TCP/IP**. Recife: Universidade Federal de Pernambuco. Dissertação de Mestrado, 1999, p.2.

INFRA, Revista Infra Magazine 10. **Monitorando o ambiente de TI:** Conheça as soluções Cacti, Wireshark e Splunk, 2010.

IsF. **Idiomas sem fronteiras.** 2017. Disponível em: <http://isf.mec.gov.br/> Acesso em 06 jun 2019.

KAMIENSKI, Carlos Alberto; SADOK, Djamel. Qualidade de serviço na internet. **minicurso, 18o SBRC, Belo Horizonte/MG**, 2000.

KUROSE, James F.; ROSS, Keith W. **Redes de Computadores e a Internet: Uma Abordagem Top-down**. São Paulo: Pearson Addison Wesley, 2006.

LAKATOS, Eva Maria; MARCONIi, Maria de Andrade. **Fundamentos de metodologia científica.** Marina de Andrade Marconi, Eva Maria Lakatos. - 5. ed. - São Paulo : Atlas 2003.

LEE, Jin-Shyan; YU-WEI SU; CHUNG-CHOU SHEN. Um estudo comparativo de protocolos sem fio: Bluetooth, UWB, ZigBee e Wi-Fi. **Sociedade de eletrônica industrial** , v. 5, p. 46-51, 2007.

LESSA, Felipe Almeida. O protocolo WEP: Sigilo contra Acidentes. **Universidade de Brasılia**, 2009. Disponível em: < https://cic.unb.br/~pedro/trabs/lessa.pdf > Acesso em 10 maio 2019.

MEIRELLES, Adriano. **Hardware Manual Completo.** Placas de rede. Disponível em: <https://www.hardware.com.br/livros/hardware-manual/placas-rede-1.html> Acessado em 25 out 2019.

MONQUEIRO, Julio Cesar Bessa. **História das redes,** 2008. Disponível em: <https://www.hardware.com.br/tutoriais/historia-redes/> Acesso em 27 maio 2019.

NAKAMURA, Emilio Tissato; DE GEUS, Paulo Lício. **Segurança de redes em ambientes cooperativos**. Novatec Editora, 2007.

SANDERS, Chris. **Análise prática de pacotes: Usando o Wireshark para resolver problemas de rede do mundo real** . Traduzido, 09/08/2010.

Sniffers. **O que é um sniffer?.** Disponível em: <https://www.avast.com/pt-br/c-sniffer> Acesso em 20 maio 2019.

SOUSA, Lindeberg Barros. Redes de computadores. **Dados Voz e Imagem**, 2009.

TANENBAUM, Andrew S.. **Redes de Computadores**: tradução [ds 3. ed. original] Insight Serviços de Informática. Rio de Janeiro: Campus, 1997.

VERGARA, Sylvia Constant. **Projetos e relatórios de pesquisa em administração**. 2 ed. São Paulo : Editora Atlas. S.A. 1998.

VIRTUALBOX. **Bem-vindo ao VirtualBox.org**. Disponível em: <https://www.virtualbox.org/> Acessado em 23 out 2019.