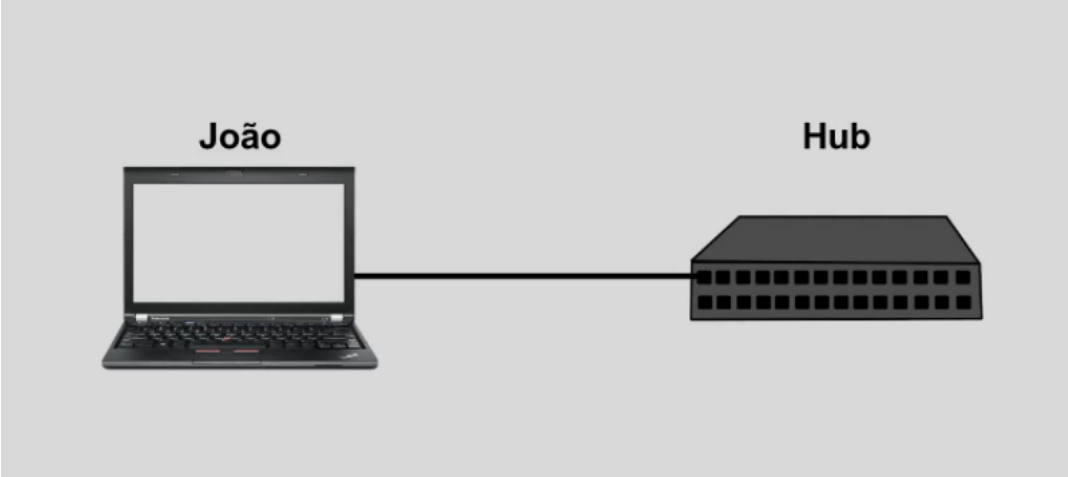
02

Hub wireshark

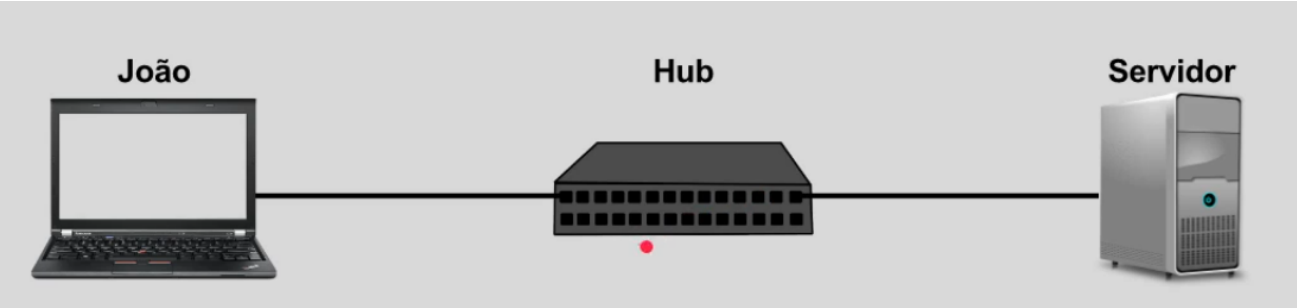
Transcrição

No curso de redes (https://cursos.alura.com.br/course/redes-introducao), havíamos conversado sobre o funcionamento de alguns equipamentos. Começamos pelo hub, que é usado para interconectar dispositivos finais. Vamos lembrar como ele trabalha?

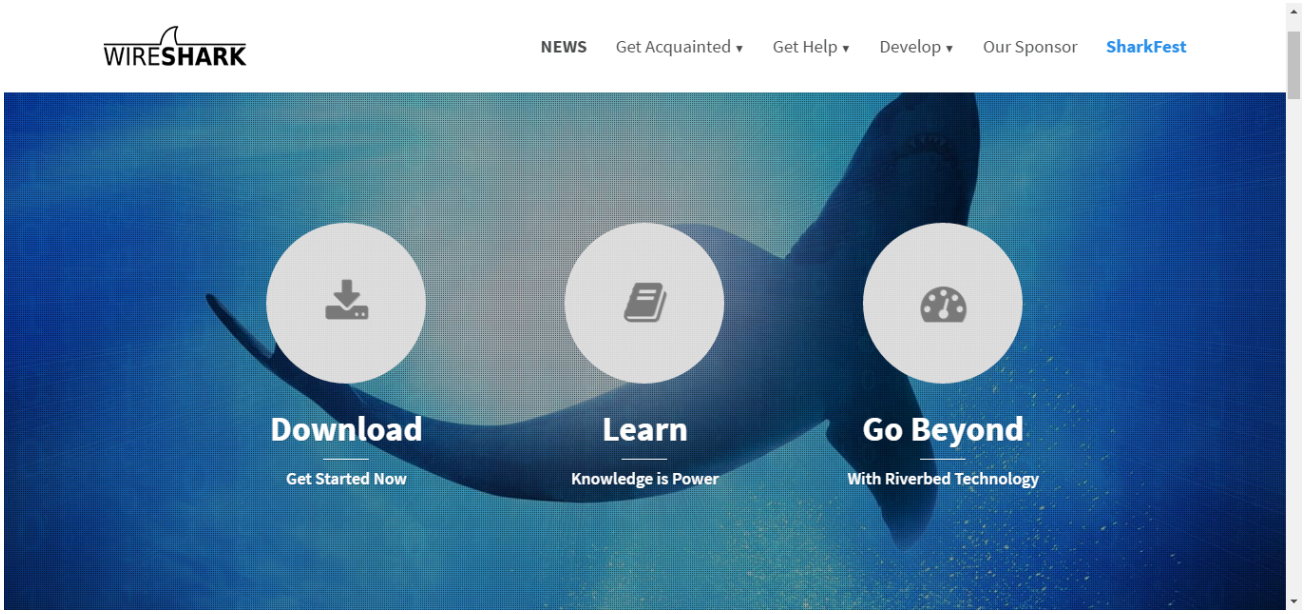
Suponha que tenhamos João com o seu computador, e que este computador está conectado a um hub.



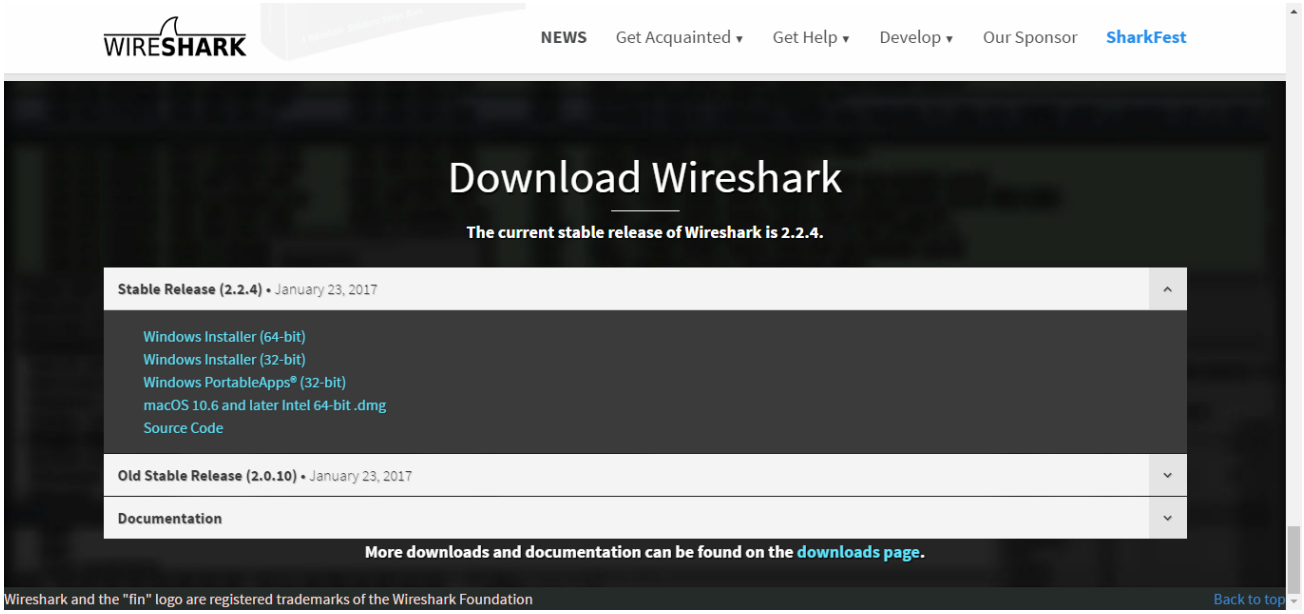
No curso de redes, fizemos algumas simulações de acesso a algumas páginas da internet, usando um servidor. No entanto, o hub tem uma limitação: ele não consegue identificar qual equipamento está conectado em cada porta.



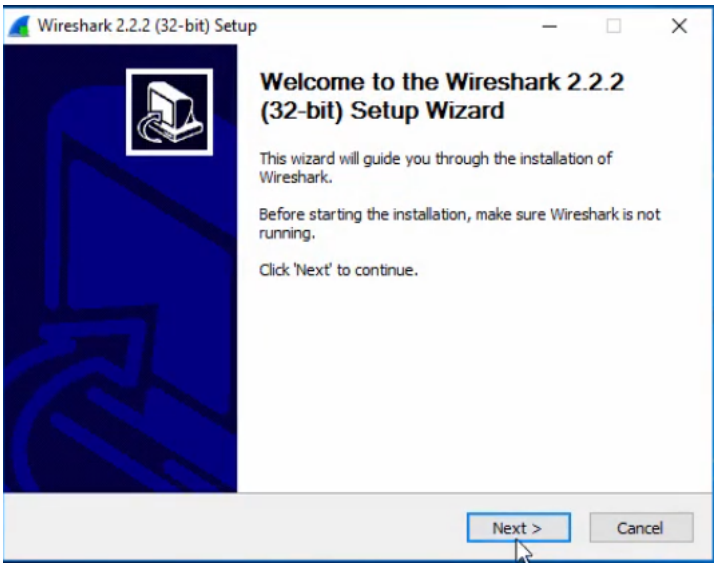
Ele não sabe que o computador do João está à esquerda e o servidor à direita. Sabendo como o hub trabalha, o hacker irá conectar o computador dele e usar programas que analisam tráfego e os protocolos que estão passando pela rede. O seu objetivo é descobrir o que João, que agora se tornou uma vítima, está acessando. O programa usado pelo hacker para colher esses dados é o Wireshark (https://www.wireshark.org/). Você pode baixá-lo e fazer uso do mesmo neste curso.



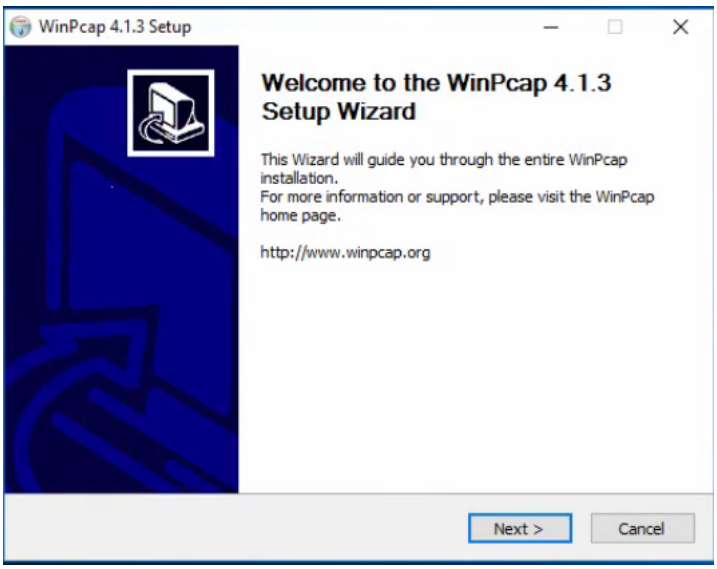
Clicando em `Download`, você é redirecionado para uma página que te permite escolher o seu sistema operacional.



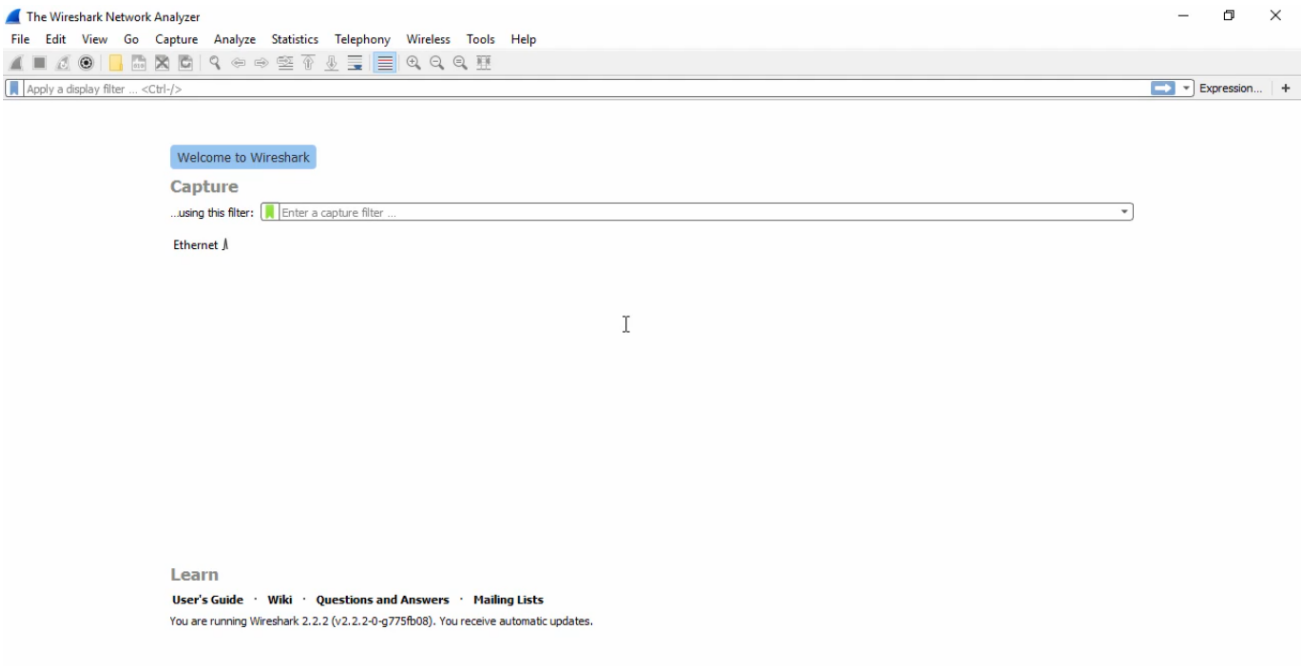
Basta clicar no que corresponde a sua máquina e o download se iniciará automaticamente. A instalação desse programa é bem imediata, basta abrir o instalador e clicar em `Next` e `I Agree` até que a instalação peça mais autorizações.



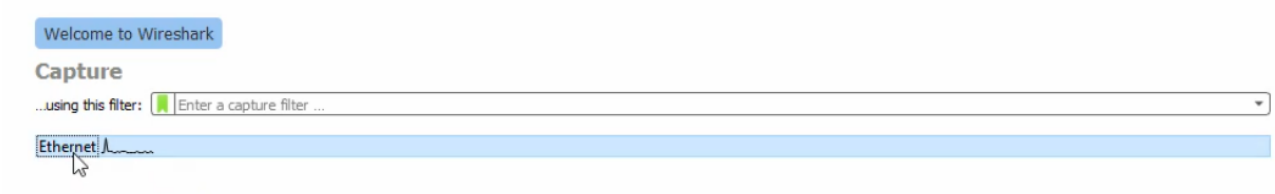
Logo a seguir abre-se o Wizard de instalação do WinPcap, que é um programa que roda por trás para pegar os protocolos. Também iremos instalá-lo.

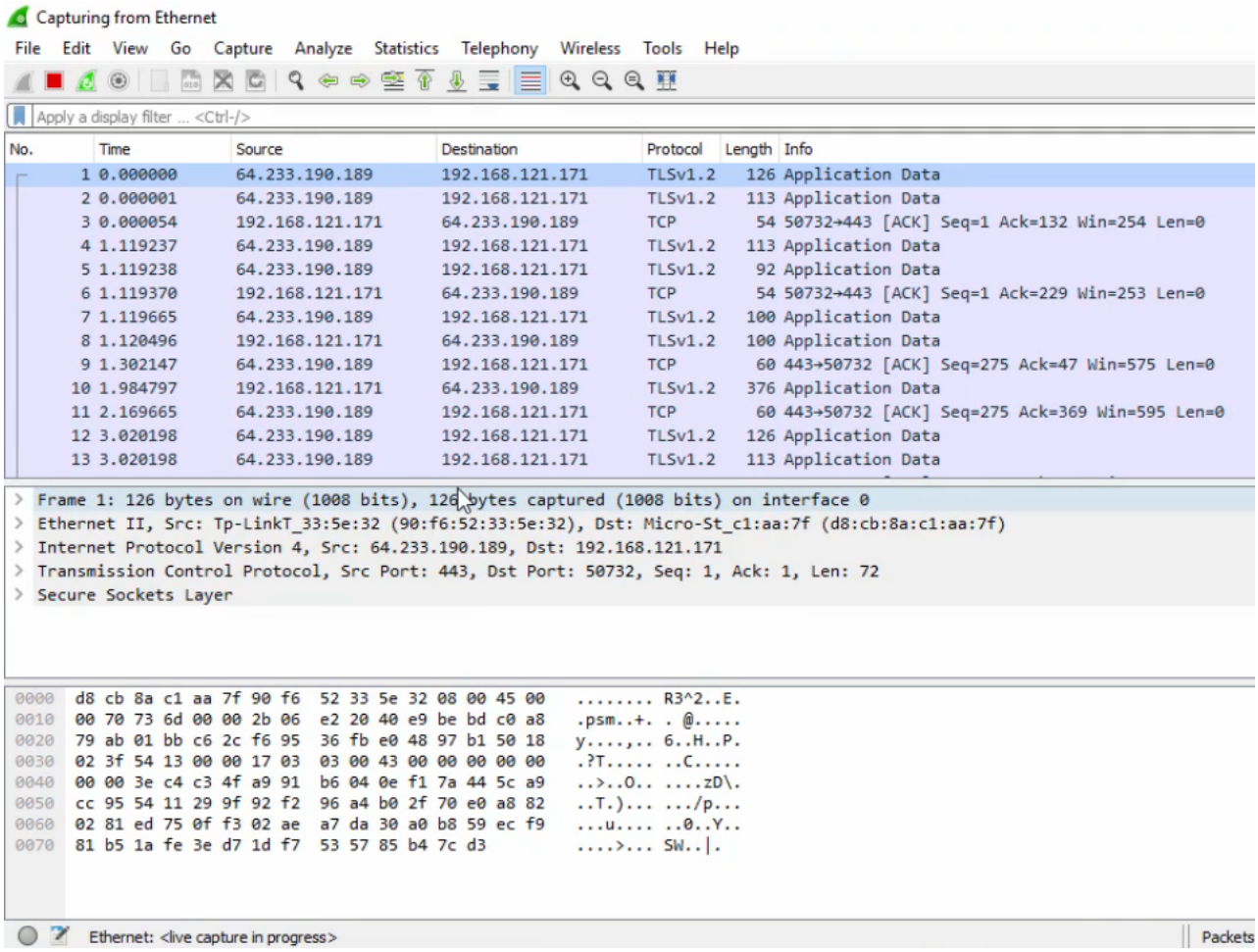


Depois de completar as instalações, abriremos o Wireshark.



Estou usando conexão cabeada aqui, por isso o programa mostra a tecnologia `Ethernet` . Na sua casa você terá outras placas de rede, e essa entrada estará de acordo com elas. Ao clicar duas vezes sobre essa placa de rede, o programa começará a nos mostrar uma série de protocolos que estão passando pela rede.

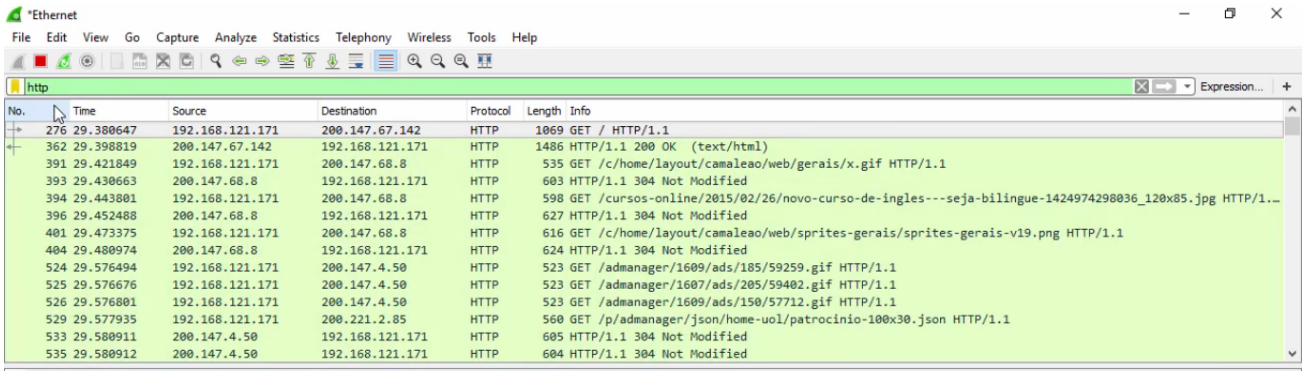




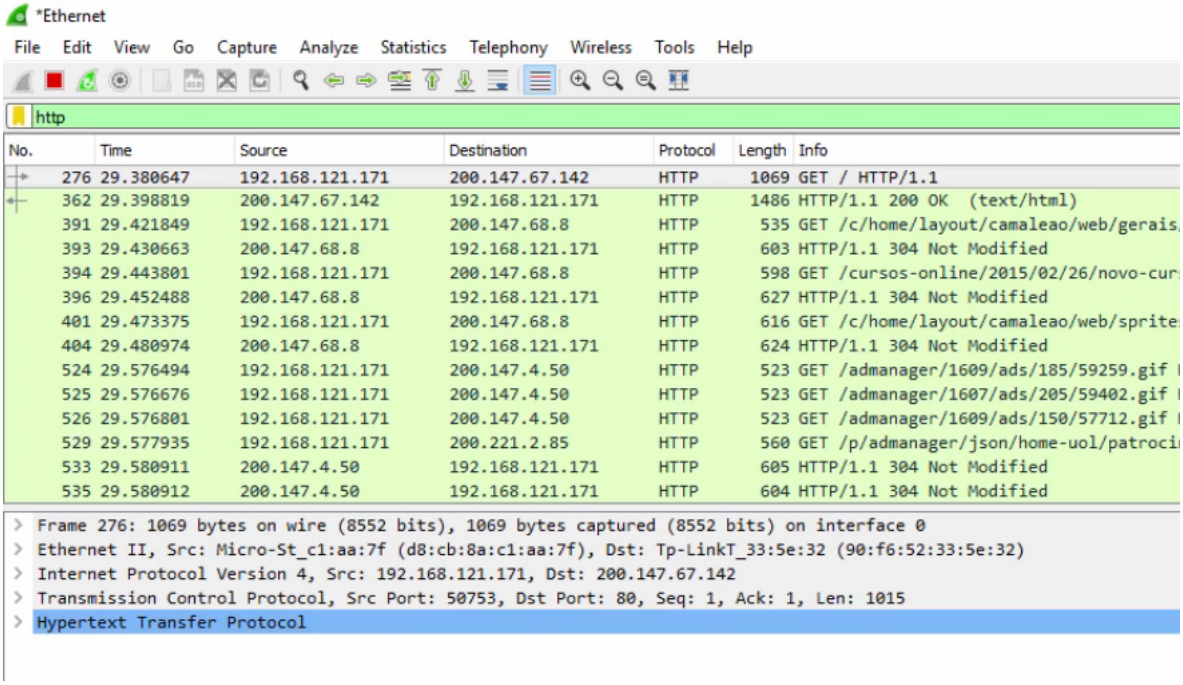
O João, a nossa vítima, quer ler algumas notícias na internet. Para isso, vai acessar o site da Uol.

Enquanto isso, o hub está passando a informação para todas as portas e o hacker está colhendo as informações de tudo o que o João acessa.

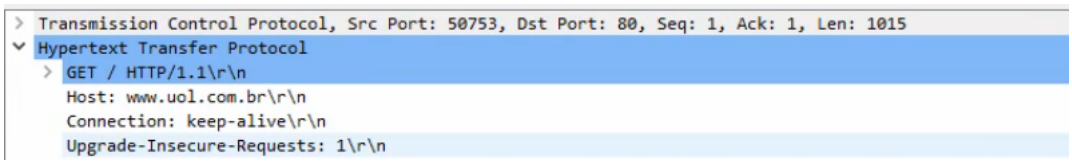
O hacker fará uma análise de protocolos no Wireshark. Faremos uma busca por protocolos HTTP, aplicando um filtro na parte superior.



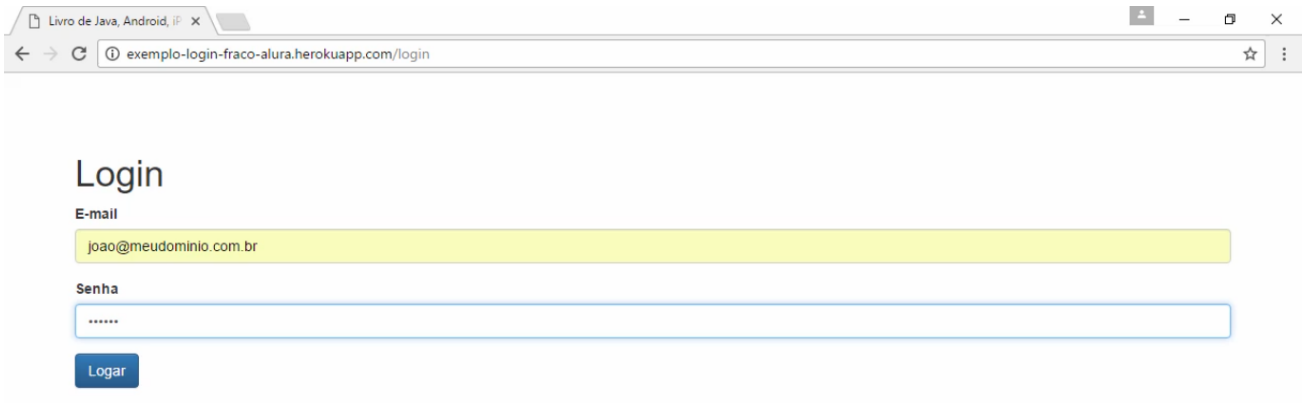
Agora vemos apenas os protocolos HTTP que foram transmitidos pelo hub. Perceba que o primeiro deles já é uma requisição `GET`. Sabemos então que algum usuário fez essa requisição para alguma página. Se olharmos embaixo, veremos que há o campo `Hypertext Transfer Protocol`.



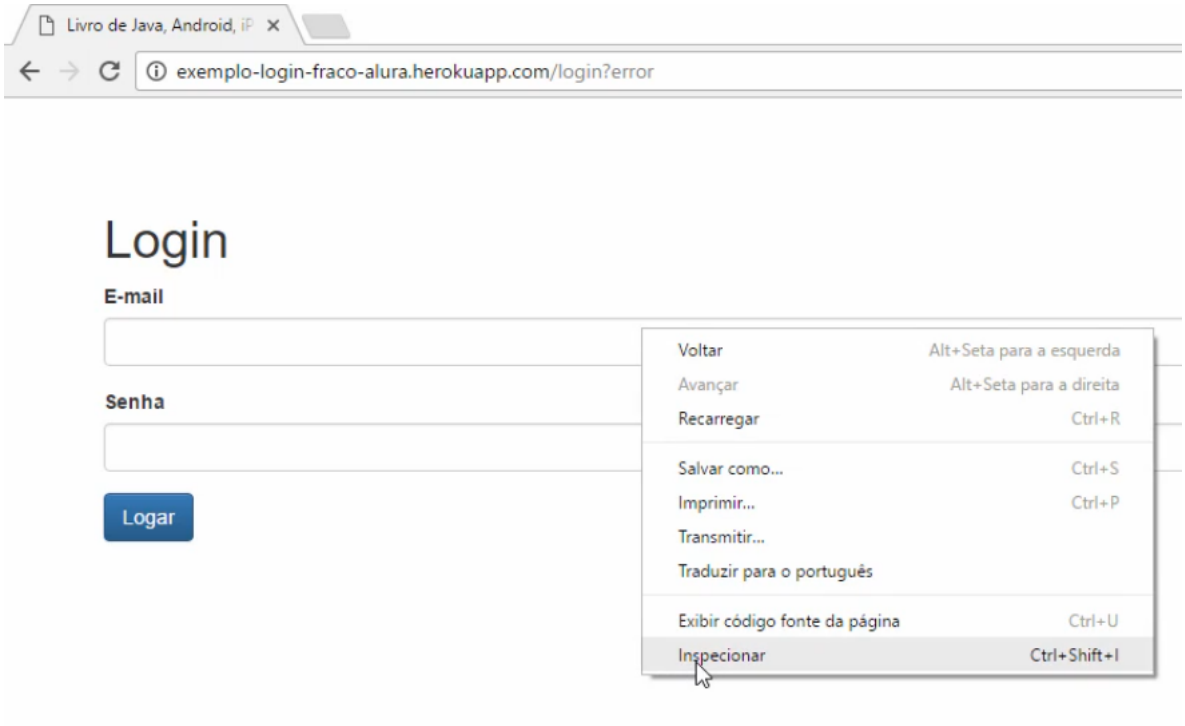
Esse é o nome oficial do HTTP, e, clicando sobre ele vemos que houve uma requisição `GET`, e que ela foi feita para o site da Uol. Assim o hacker sabe que o João acessou esse site.



E o hacker pode saber ainda mais. Suponha que o João precisa fazer cadastro em um site e se logar na parte administrativa deste site.



Quando o João se loga, o hacker está obtendo as informações. Vamos inspecionar a página para ver como está estruturado esse formulário.

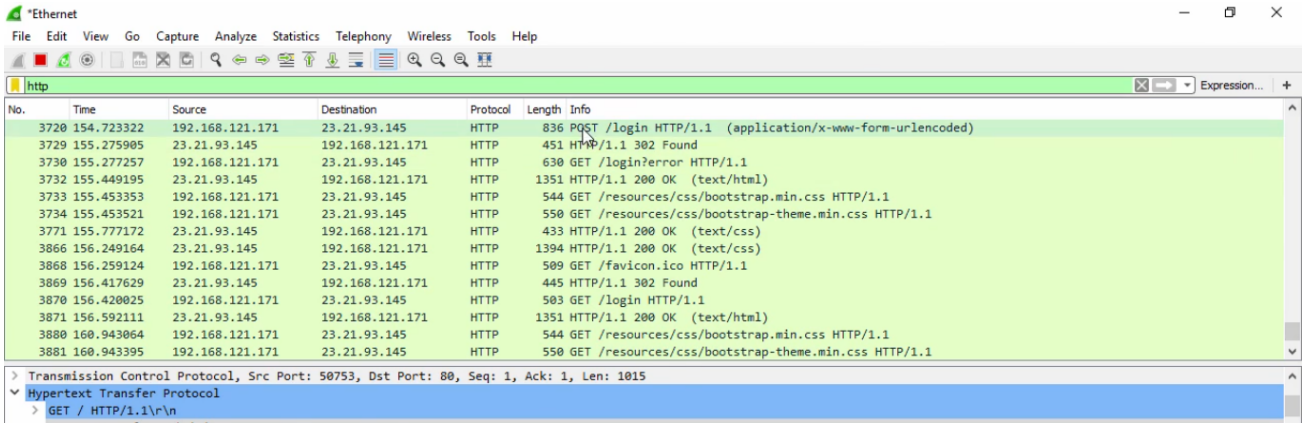


O que veremos, abrindo cada nível, é o seguinte código:

```
...<html>==</head>
  <head>...</head>
  <body>
    <div class="container">
      ::before
      <h1>Login</h1>
      <form id="command" action="/login" method"POST">
        <div class="form-group">...</div>
        <div class="form-group">...</div>
        <button type="submit" class="btn btn-primary">Logar</button>
        <div>...</div>
      </form>
      ::after
    </div>
  </body>
</html>
```

Observe que fizemos uma requisição `POST`, e é por meio dela que o usuário envia as informações para o servidor. Vamos ver se no Wireshark nós também encontramos essa requisição.

Como é uma requisição recente, vamos descer a barra de rolagem para encontrar essa requisição.

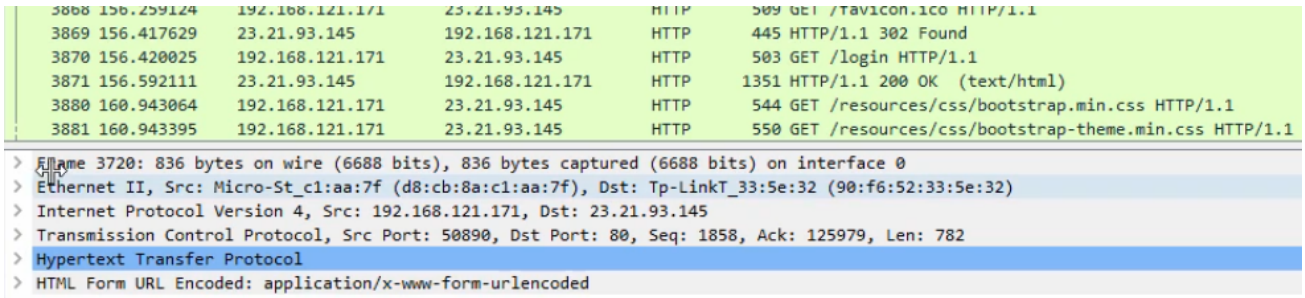


Clicando sobre ele, veremos mais detalhes no campo inferior.

```
Hipertext Transfer Protocol
  POST /login HTTP/1.1\r\n
  Host: exemplo-login-fraco-alura.herokuapp.com\r\n
  Connection: keep-alive\r\n
  Content-Lenght: 92\r\n
  Cache-Control: max-age=0\r\n
  Origin: http://exemplo-login-fraco-alura.herokuapp.com\r\n
  Upgrade-Insecure-Requests 1\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) (
  Content-Type: application/x-www-form-urlencoded\r\n
  Accept: text/html, application/xhtml+xml, application/xml;q=0.9, image/webp,*/*;q=0.8\r\n
  Referer: http://exemplo-login-fraco-alura.herokuapp.com/login\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: pt-BR, pt;q=0.8,en-US;q=0.6,en;q=0.4\r\n
  Cookie: JSESSIONID=DDE16B48A56B0C958F2FE8413BC16F8F\r\n
  \r\n
```

Vemos que houve uma requisição `POST` e a página em que o João a fez era `exemplo-login-fraco-alura.herokuapp.com\r\n`. Mas não há nenhuma informação tão consistente. Será que o hacker teria acesso à senha ou ao login do João?

Embaixo do campo `Hypertext Transfer Protocol`, há o campo `HTML Form URL Encoded: application/x-www-form-urlencoded`.

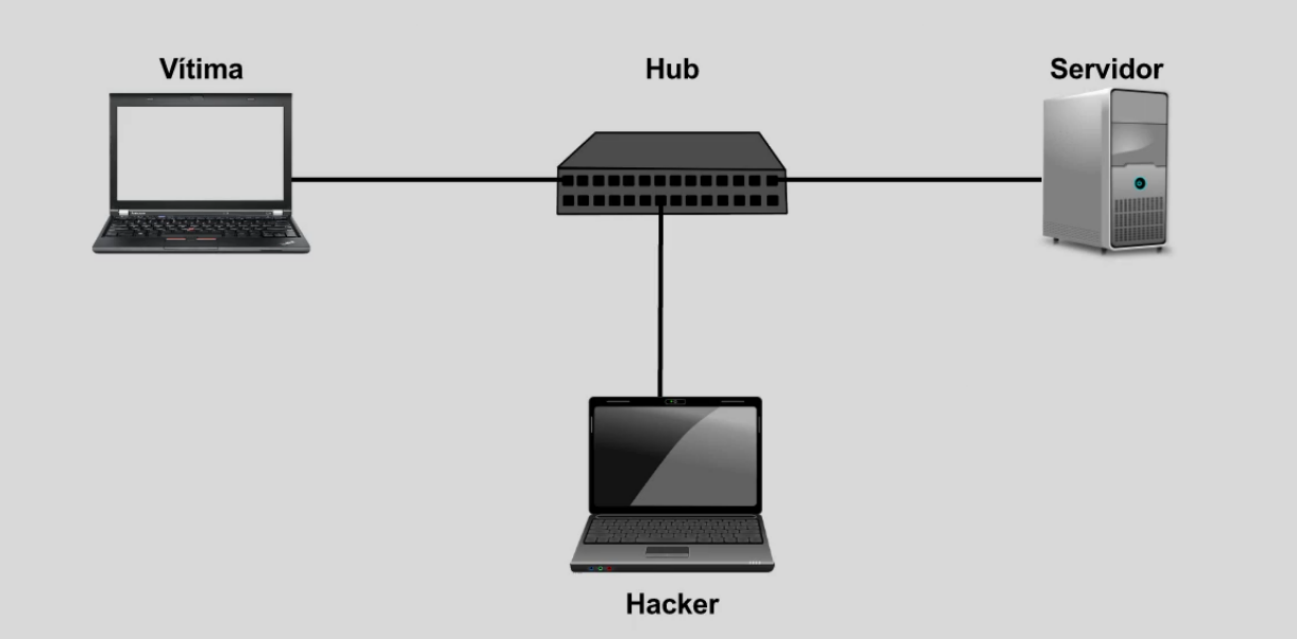


Quando o abrirmos, teremos:

```
Form item: "username" = "joao@meudominio.com.br"
Form item: "password" = "123456"
Form item: "_csrf" = "469bde26-87c6-4e9c-82d3-ae2c61a41fe3"
```

Conseguimos ver o email e a senha que o João digitou no cadastro do site.

Percebemos uma limitação do hub, que é essa vulnerabilidade do usuário. O hacker consegue fazer essa análise dos protocolos e ver as informações que os outros usuários estão transmitindo. Além disso, há também um problema de lentidão.



Como o hub não consegue identificar que máquina está em cada porta, ele passa essas informações para todas as portas. E por isso, temos a lentidão.

Existe um equipamento que é praticamente uma evolução do hub, e que veio suprir as limitações que mencionamos. Essa tecnologia é o switch. Em breve veremos como ele trabalha. Até lá!