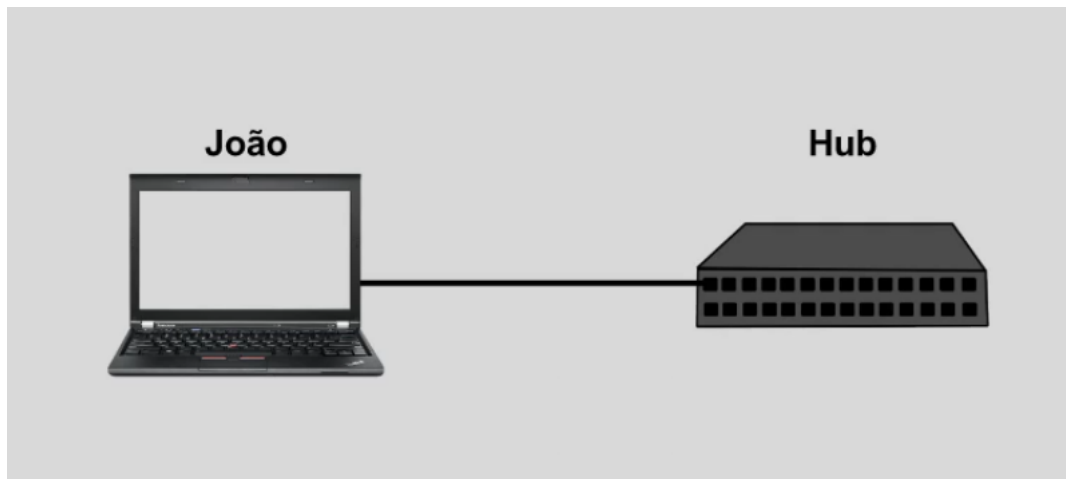


Hub wireshark

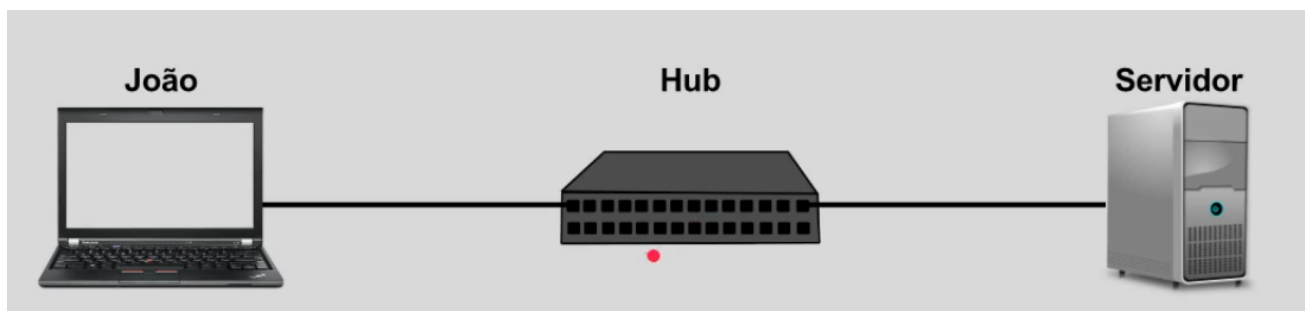
Transcrição

No curso de [redes](https://cursos.alura.com.br/course/redes-introducao) (<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?

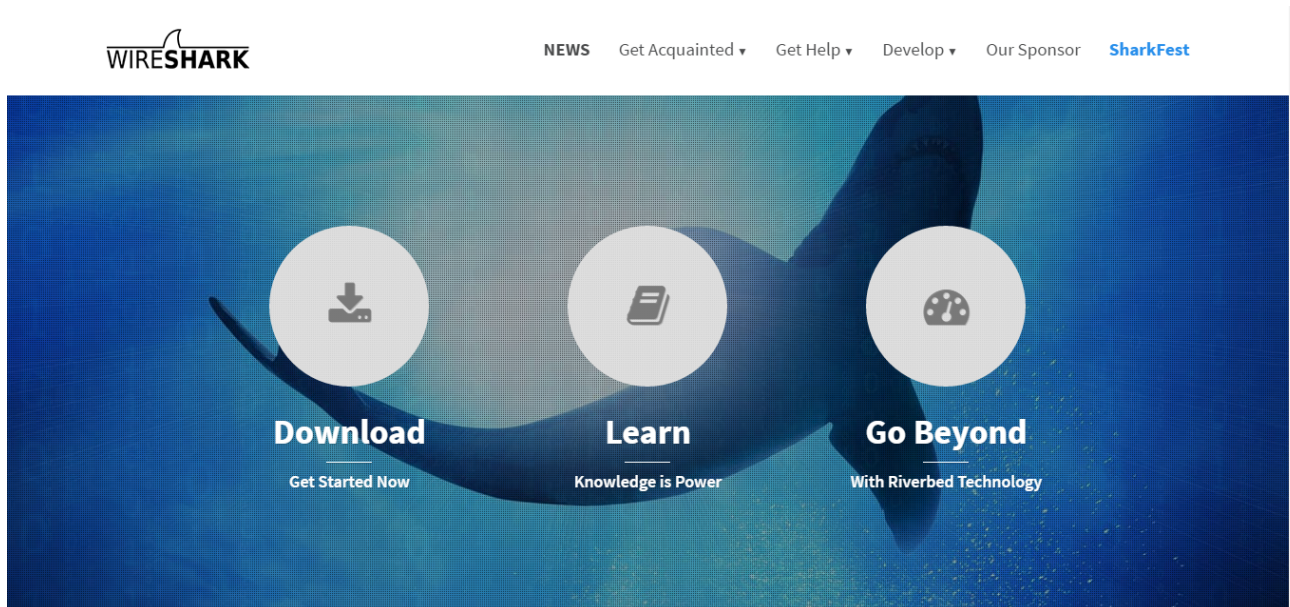
Suponha que tenhamos João com o seu computador, e que este computador está conectado a um hub.



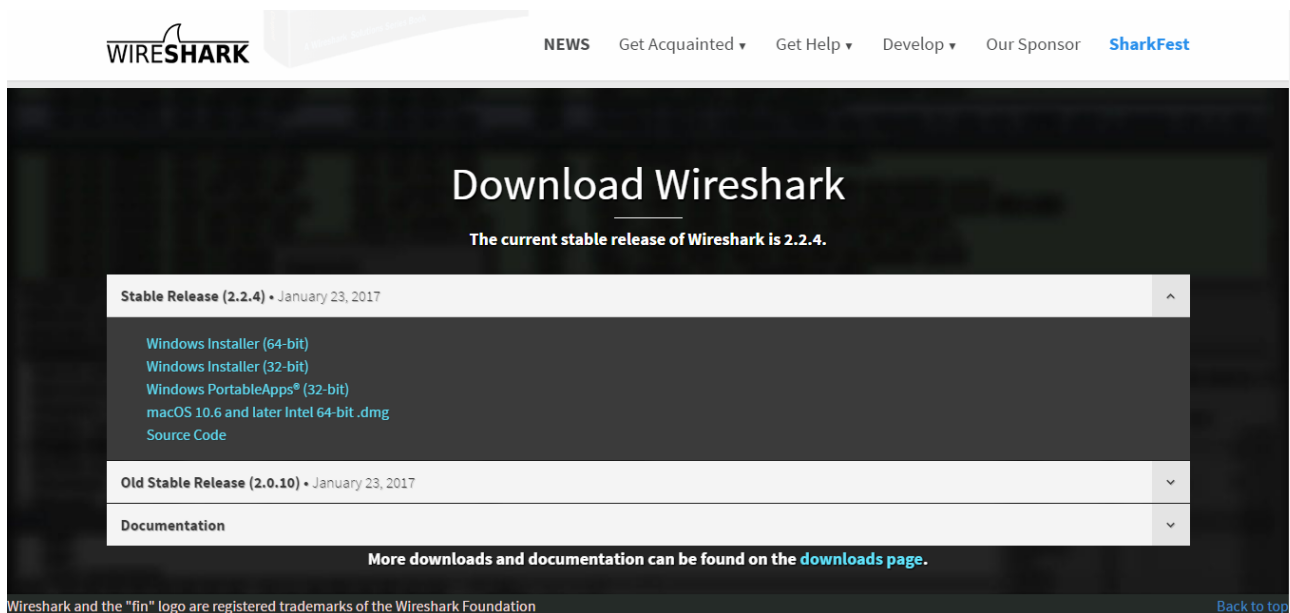
No curso de redes, fizemos algumas simulações de acesso a algumas páginas da internet, usando um servidor. No entanto, o hub tem uma limitação: ele não consegue identificar qual equipamento está conectado em cada porta.



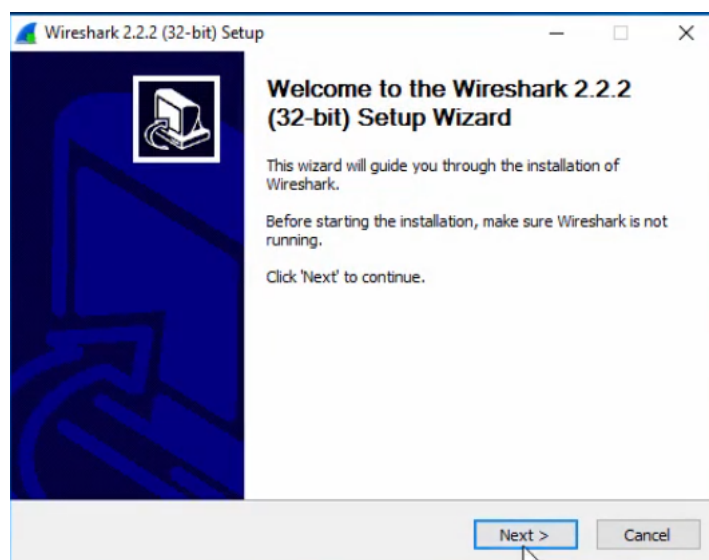
Ele não sabe que o computador do João está à esquerda e o servidor à direita. Sabendo como o hub trabalha, o hacker irá conectar o computador dele e usar programas que analisam tráfego e os protocolos que estão passando pela rede. O seu objetivo é descobrir o que João, que agora se tornou uma vítima,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



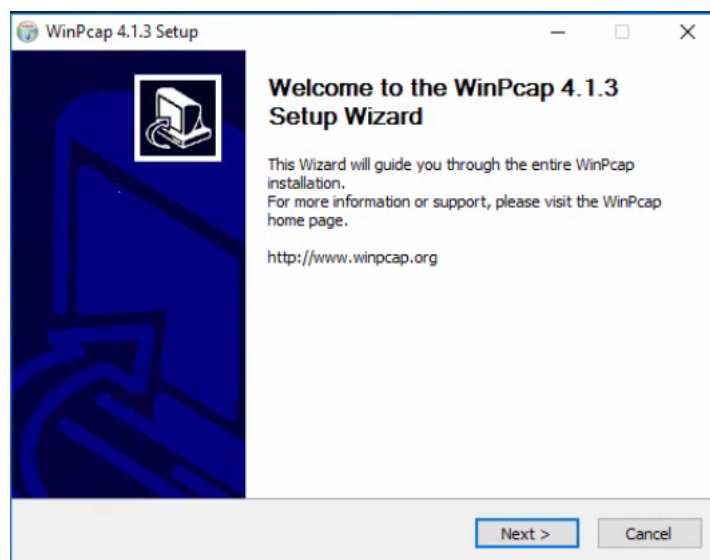
Clicando em **Download**, você é redirecionado para uma página que te permite escolher o seu sistema operacional.



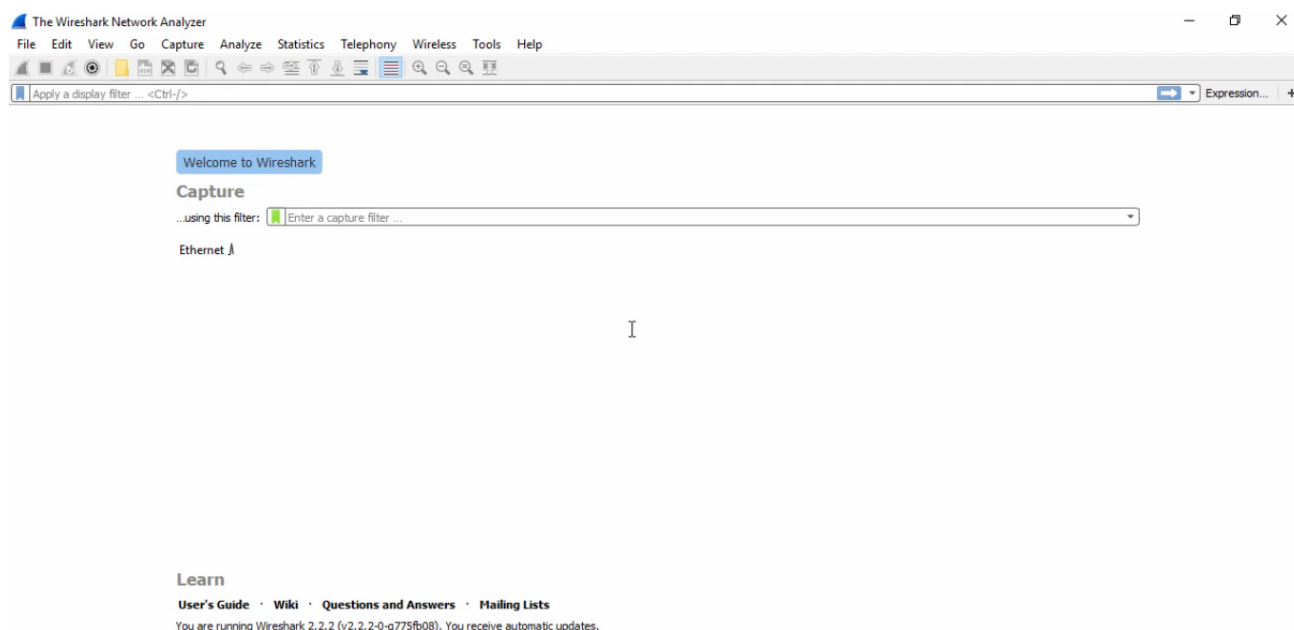
Basta clicar no que corresponde a sua máquina e o download se iniciará automaticamente. A instalação desse programa é bem imediata, basta abrir o instalador e clicar em **Next** e **I Agree** até que a instalação peça mais autorizações.



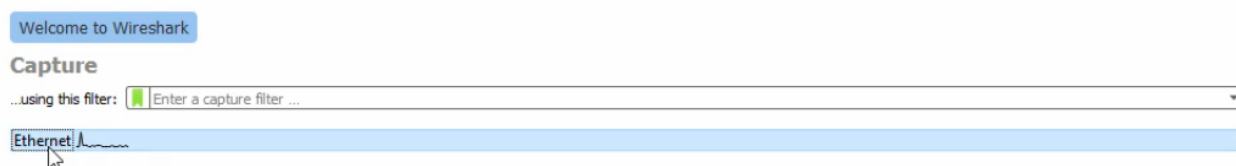
Logo a seguir abre-se o Wizard de instalação do WinPcap, que é um programa que roda por trás para pegar os protocolos. Também iremos instalá-lo.



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.



Capturing from Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	64.233.190.189	192.168.121.171	TLSv1.2	126	Application Data
2	0.000001	64.233.190.189	192.168.121.171	TLSv1.2	113	Application Data
3	0.000054	192.168.121.171	64.233.190.189	TCP	54	50732→443 [ACK] Seq=1 Ack=132 Win=254 Len=0
4	1.119237	64.233.190.189	192.168.121.171	TLSv1.2	113	Application Data
5	1.119238	64.233.190.189	192.168.121.171	TLSv1.2	92	Application Data
6	1.119370	192.168.121.171	64.233.190.189	TCP	54	50732→443 [ACK] Seq=1 Ack=229 Win=253 Len=0
7	1.119665	64.233.190.189	192.168.121.171	TLSv1.2	100	Application Data
8	1.120496	192.168.121.171	64.233.190.189	TLSv1.2	100	Application Data
9	1.302147	64.233.190.189	192.168.121.171	TCP	60	443→50732 [ACK] Seq=275 Ack=47 Win=575 Len=0
10	1.984797	192.168.121.171	64.233.190.189	TLSv1.2	376	Application Data
11	2.169665	64.233.190.189	192.168.121.171	TCP	60	443→50732 [ACK] Seq=275 Ack=369 Win=595 Len=0
12	3.020198	64.233.190.189	192.168.121.171	TLSv1.2	126	Application Data
13	3.020198	64.233.190.189	192.168.121.171	TLSv1.2	113	Application Data

> Frame 1: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface 0

> Ethernet II, Src: Tp-LinkT_33:5e:32 (90:f6:52:33:5e:32), Dst: Micro-St_c1:aa:7f (d8:cb:8a:c1:aa:7f)

> Internet Protocol Version 4, Src: 64.233.190.189, Dst: 192.168.121.171

> Transmission Control Protocol, Src Port: 443, Dst Port: 50732, Seq: 1, Ack: 1, Len: 72

> Secure Sockets Layer

```

0000  d8 cb 8a c1 aa 7f 90 f6 52 33 5e 32 08 00 45 00 ..... R3^2..E.
0010  00 70 73 6d 00 00 2b 06 e2 20 40 e9 be bd c0 a8 .psm..+. . @....
0020  79 ab 01 bb c6 2c f6 95 36 fb e0 48 97 b1 50 18 y..... 6..H..P.
0030  02 3f 54 13 00 00 17 03 03 00 43 00 00 00 00 00 .?T..... .C.....
0040  00 00 3e c4 c3 4f a9 91 b6 04 0e f1 7a 44 5c a9 ..>..O. ....zD\
0050  cc 95 54 11 29 9f 92 f2 96 a4 b0 2f 70 e0 a8 82 ..T.)... ../p...
0060  02 81 ed 75 0f f3 02 ae a7 da 30 a0 b8 59 ec f9 ...u.... ..Y..
0070  81 b5 1a fe 3e d7 1d f7 53 57 85 b4 7c d3 ....>... SW..|.

```

Ethernet: <live capture in progress> Packets

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

BOL UOL HOST PAGSEGURO CURSOS + PRODUTOS
 06 de Dezembro de 2016 Dólar - R\$ 3,454

UOL MAIL e-mail seguro
 Usuário Senha Entrar
 Busque na web, no UOL ou
 Ainda não tem um e-mail UOL? Crie o seu

ASSINE SAC BATE-PAPO NOTÍCIAS CARROS ECONOMIA FOLHA ESPORTE ENTRETENIMENTO TV E FAMOSOS JOGOS ESTILO

Programa Internacional de Avaliação de Estudantes
 Maioria dos alunos brasileiros erra ao fazer contas e não entende o que lê
 Brasil tem 6ª pior nota em matemática entre 76 regiões avaliadas

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.

No.	Time	Source	Destination	Protocol	Length	Info
276	29.380647	192.168.121.171	200.147.67.142	HTTP	1069	GET / HTTP/1.1
362	29.398819	200.147.67.142	192.168.121.171	HTTP	1486	HTTP/1.1 200 OK (text/html)
391	29.421849	192.168.121.171	200.147.68.8	HTTP	535	GET /c/home/layout/camaleao/web/gerais/x.gif HTTP/1.1
393	29.430663	200.147.68.8	192.168.121.171	HTTP	603	HTTP/1.1 304 Not Modified
394	29.443801	192.168.121.171	200.147.68.8	HTTP	598	GET /cursos-online/2015/02/26/novo-curso-de-ingles---seja-bilingue-1424974298036_120x85.jpg HTTP/1.1
396	29.452488	200.147.68.8	192.168.121.171	HTTP	627	HTTP/1.1 304 Not Modified
401	29.473375	192.168.121.171	200.147.68.8	HTTP	616	GET /c/home/layout/camaleao/web/sprites-gerais/sprites-gerais-v19.png HTTP/1.1
404	29.480974	200.147.68.8	192.168.121.171	HTTP	624	HTTP/1.1 304 Not Modified
524	29.576494	192.168.121.171	200.147.4.50	HTTP	523	GET /admanager/1609/ads/185/59259.gif HTTP/1.1
525	29.576676	192.168.121.171	200.147.4.50	HTTP	523	GET /admanager/1607/ads/205/59402.gif HTTP/1.1
526	29.576801	192.168.121.171	200.147.4.50	HTTP	523	GET /admanager/1609/ads/150/57712.gif HTTP/1.1
529	29.577935	192.168.121.171	200.221.2.85	HTTP	560	GET /p/admanager/json/home-uol/patrocinio-100x30.json HTTP/1.1
533	29.580911	200.147.4.50	192.168.121.171	HTTP	605	HTTP/1.1 304 Not Modified
535	29.580912	200.147.4.50	192.168.121.171	HTTP	604	HTTP/1.1 304 Not Modified

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.

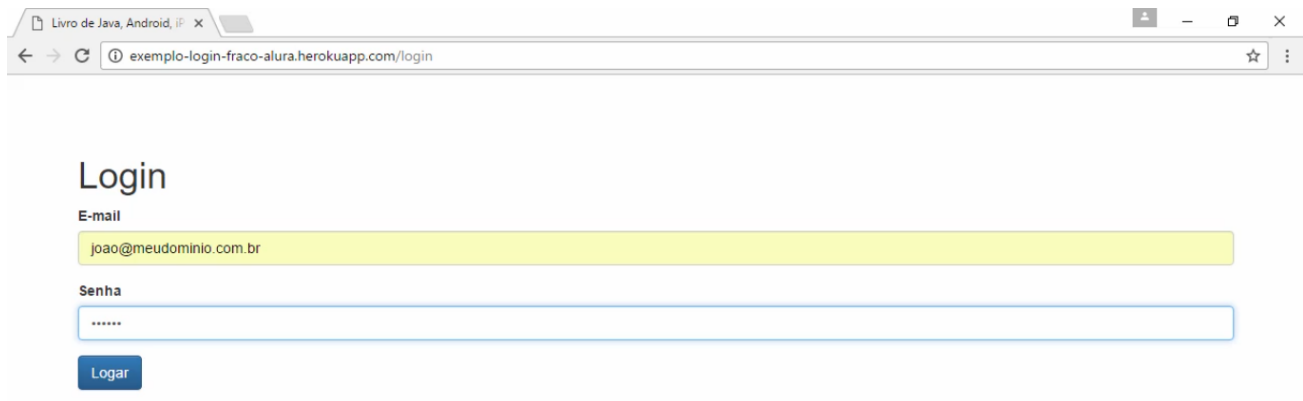
No.	Time	Source	Destination	Protocol	Length	Info
276	29.380647	192.168.121.171	200.147.67.142	HTTP	1069	GET / HTTP/1.1
362	29.398819	200.147.67.142	192.168.121.171	HTTP	1486	HTTP/1.1 200 OK (text/html)
391	29.421849	192.168.121.171	200.147.68.8	HTTP	535	GET /c/home/layout/camaleao/web/gerais/x.gif HTTP/1.1
393	29.430663	200.147.68.8	192.168.121.171	HTTP	603	HTTP/1.1 304 Not Modified
394	29.443801	192.168.121.171	200.147.68.8	HTTP	598	GET /cursos-online/2015/02/26/novo-curso-de-ingles---seja-bilingue-1424974298036_120x85.jpg HTTP/1.1
396	29.452488	200.147.68.8	192.168.121.171	HTTP	627	HTTP/1.1 304 Not Modified
401	29.473375	192.168.121.171	200.147.68.8	HTTP	616	GET /c/home/layout/camaleao/web/sprites-gerais/sprites-gerais-v19.png HTTP/1.1
404	29.480974	200.147.68.8	192.168.121.171	HTTP	624	HTTP/1.1 304 Not Modified
524	29.576494	192.168.121.171	200.147.4.50	HTTP	523	GET /admanager/1609/ads/185/59259.gif HTTP/1.1
525	29.576676	192.168.121.171	200.147.4.50	HTTP	523	GET /admanager/1607/ads/205/59402.gif HTTP/1.1
526	29.576801	192.168.121.171	200.147.4.50	HTTP	523	GET /admanager/1609/ads/150/57712.gif HTTP/1.1
529	29.577935	192.168.121.171	200.221.2.85	HTTP	560	GET /p/admanager/json/home-uol/patrocinio-100x30.json HTTP/1.1
533	29.580911	200.147.4.50	192.168.121.171	HTTP	605	HTTP/1.1 304 Not Modified
535	29.580912	200.147.4.50	192.168.121.171	HTTP	604	HTTP/1.1 304 Not Modified

> Frame 276: 1069 bytes on wire (8552 bits), 1069 bytes captured (8552 bits) on interface 0 > Ethernet II, Src: Micro-St_c1:aa:7f (d8:cb:8a:c1:aa:7f), Dst: Tp-LinkT_33:5e:32 (90:f6:52:33:5e:32) > Internet Protocol Version 4, Src: 192.168.121.171, Dst: 200.147.67.142 > Transmission Control Protocol, Src Port: 50753, Dst Port: 80, Seq: 1, Ack: 1, Len: 1015 > Hypertext Transfer Protocol						
---	--	--	--	--	--	--

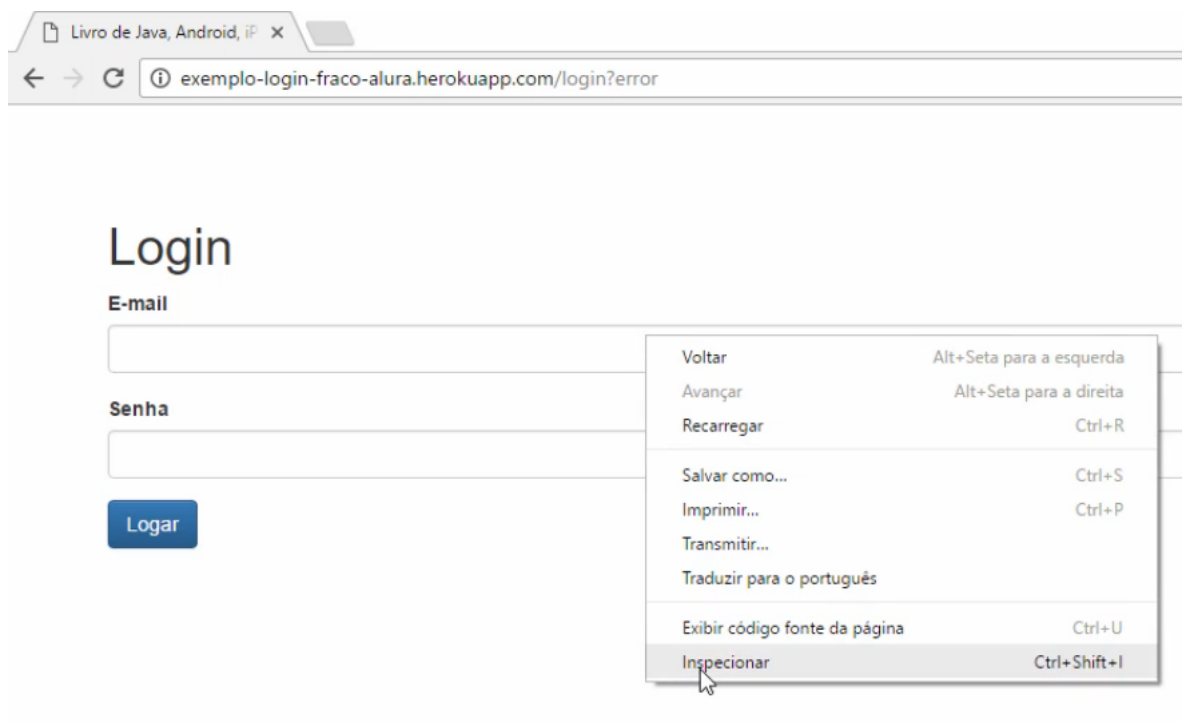
Esse é o nome oficial do HTTP, e, clicando sobre ele vemos que houve uma requisição GET, e que ela foi feita para o site da Uol. Assim o hacker sabe que o João acessou esse site.

> Transmission Control Protocol, Src Port: 50753, Dst Port: 80, Seq: 1, Ack: 1, Len: 1015	
> Hypertext Transfer Protocol	
> GET / HTTP/1.1\r\n Host: www.uol.com.br\r\n Connection: keep-alive\r\n Upgrade-Insecure-Requests: 1\r\n	

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.

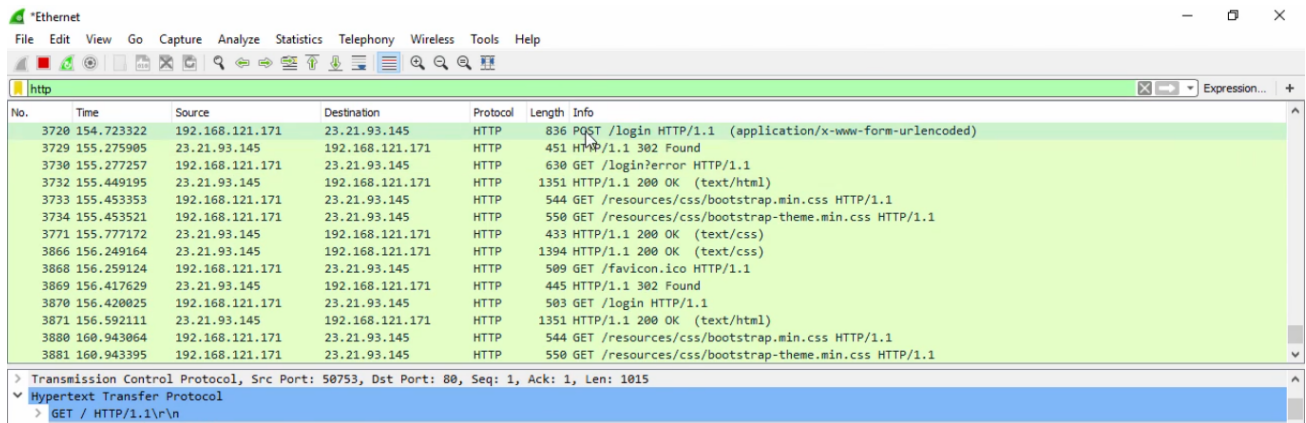


O que veremos, abrindo cada nível, é o seguinte código:

```
...<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.



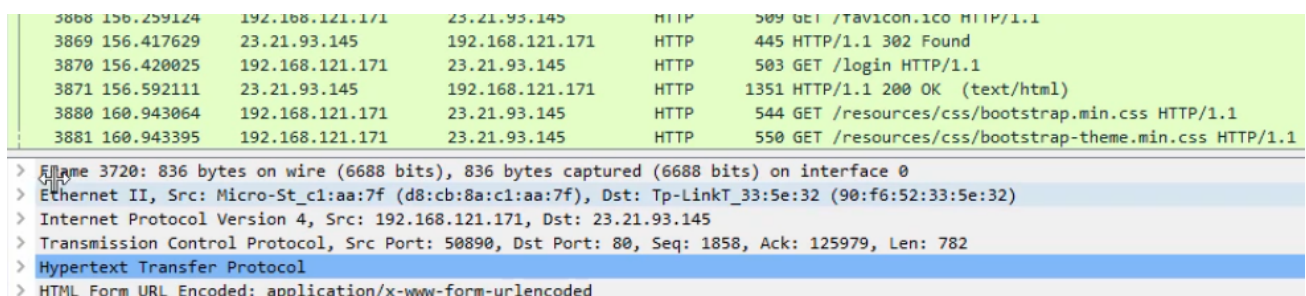
Clicando sobre ele, veremos mais detalhes no campo inferior.

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`.

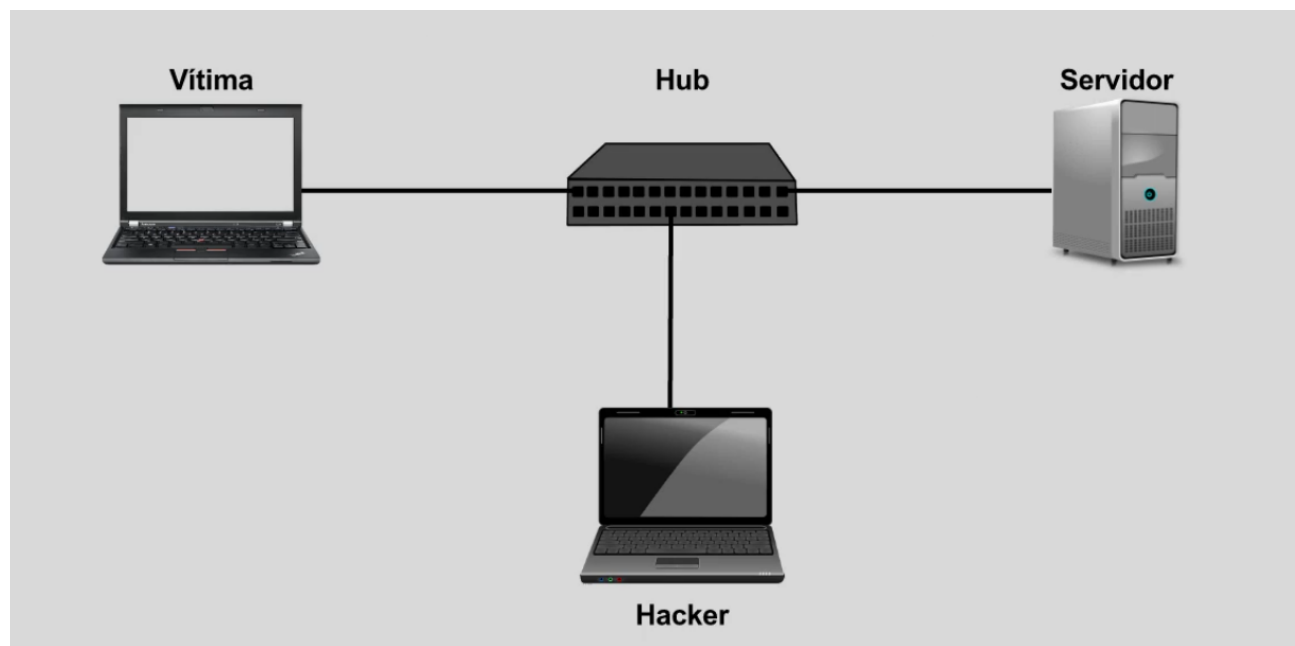


Quando o abrirmos, teremos:

```
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!