

Projeto 1

Redes de Computadores

João Pedro Assunção Coutinho, 18/0019813
João Victor de Souza Calassio, 18/0033808

1

1. Introdução

O primeiro projeto da disciplina visa, através da utilização de um software 'package sniffer' observar o funcionamento de um programa que estabeleça uma conexão do tipo cliente-servidor, de forma que seja possível identificar o fluxo de pacotes por meio desse software(Wireshark é o utilizado aqui).

O programa construído pelo grupo, utiliza a API do YouTube para recuperar o título e número de visualizações de um vídeo, utilizando seu link.

2. Execução

O programa foi feito utilizando Python 3, portanto, para a execução, é necessário em primeiro lugar, instalar o interpretador. Instruções de instalação genéricas podem ser encontradas na página oficial: <https://www.python.org/downloads/release/python-374/>

Após a instalação do interpretador Python 3, é necessário a instalação da biblioteca gráfica Tkinter, e as instruções de instalação genéricas podem ser encontradas nessa página: <https://tkdocs.com/tutorial/install.html>

Após a instalação dos pré-requisitos, basta executar o programa *main.py*, utilizando o comando (caso no Linux) :

```
$ python3 main.py
```

O comportamento esperado é que seja aberta uma janela com uma caixa para a entrada do link, e um botão 'Search' que iniciará a busca. Quando o processo de busca acabar, espera-se que um texto abaixo da caixa de entrada mostre o título seguido do número de visualizações do vídeo correspondente ao link dado como input.

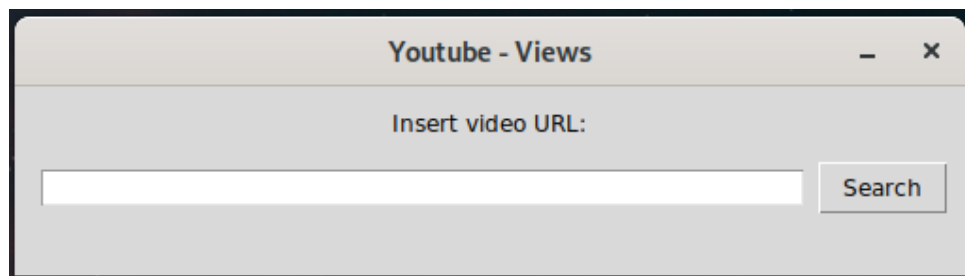


Figure 1. Janela aberta ao executar o programa.

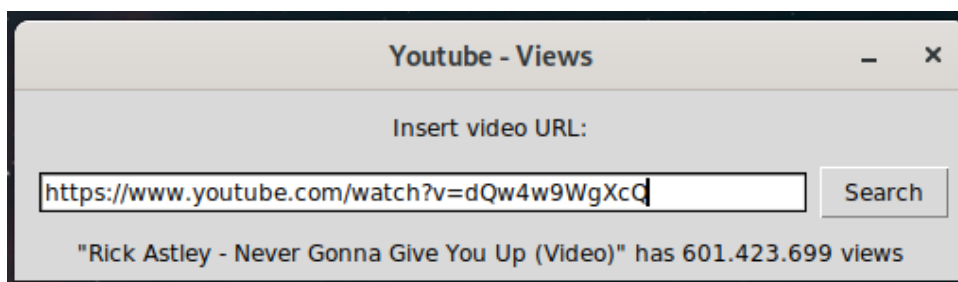


Figure 2. Indicação do título e do número de views, após buscar pelo link.

3. Análise Experimental

Por se tratar de um servidor web remoto, o grupo não precisou definir os parâmetros de configuração. No entanto, todos os requisitos de acesso simultâneo são cumpridos.

3.1. Concepção das funcionalidades do cliente

O cliente utiliza a biblioteca *socket* presente no Python 3.7.1 para realizar a conexão. Como o site alvo requer a utilização de TLS/SSL, a biblioteca SSL do Python foi adicionada.

A conexão é realizada via TCP, e foram definidos 5 segundos de tempo limite. O socket é então protegido utilizando a função *wrap_socket*, e é realizada uma requisição HTTP utilizando o método GET para receber o JSON correspondente ao video desejado.

Na resposta, são recebidos bytes até chegar o tempo limite, e os headers são descartados. O JSON é então, tratado e a resposta esperada é exibida na tela. Em caso de erro, uma mensagem é exibida no lugar.

3.2. Captura de transmissão de dados

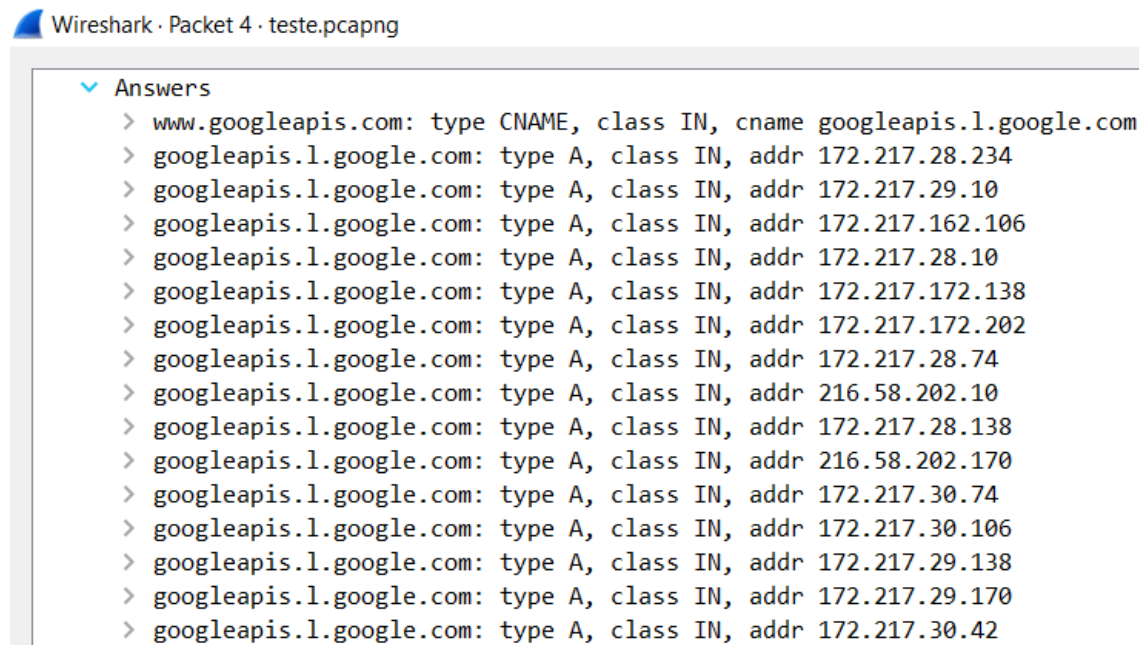
Os pacotes enviados e recebidos pelo programa foram monitorados utilizando o *packet sniffer* Wireshark.

3	18.020642854	172.29.54.230	164.41.101.11	DNS	78 Standard query 0xaa0b A www.googleapis.com
4	18.072192046	164.41.101.11	172.29.54.230	DNS	352 Standard query response 0xaa0b A www.googleapis.com CNAME googleapis.l.google.com A 172.29.54.230
5	18.072917528	172.29.54.230	172.217.28.234	TCP	74 38750 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2068906215 TSecr=0 WS=0
6	18.122406642	172.217.28.234	172.29.54.230	TCP	74 443 → 38750 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1386 SACK_PERM=1 TSval=294718686 TSecr=0
7	18.122506636	172.29.54.230	172.217.28.234	TCP	66 38750 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2068906265 TSecr=294718686
8	18.123080169	172.29.54.230	172.217.28.234	TLSv1.3	583 Client Hello
9	18.130763077	172.217.28.234	172.29.54.230	TCP	66 443 → 38750 [ACK] Seq=1 Ack=518 Win=15616 Len=0 TSval=294718689 TSecr=2068906266
10	18.208845281	172.217.28.234	172.29.54.230	TLSv1.3	1440 Server Hello, Change Cipher Spec
11	18.208907241	172.29.54.230	172.217.28.234	TCP	66 38750 → 443 [ACK] Seq=518 Ack=1375 Win=64128 Len=0 TSval=2068906351 TSecr=294718697
12	18.208961459	172.217.28.234	172.29.54.230	TLSv1.3	78 Application Data
13	18.208990245	172.29.54.230	172.217.28.234	TCP	66 38750 → 443 [ACK] Seq=518 Ack=1387 Win=64128 Len=0 TSval=2068906351 TSecr=294718697
14	18.210253093	172.29.54.230	172.217.28.234	TLSv1.3	146 Change Cipher Spec, Application Data
15	18.211718322	172.217.28.234	172.29.54.230	TCP	66 443 → 38750 [ACK] Seq=1387 Ack=598 Win=15616 Len=0 TSval=294718697 TSecr=2068906353
16	18.211792785	172.29.54.230	172.217.28.234	TLSv1.3	311 Application Data
17	18.213935732	172.217.28.234	172.29.54.230	TCP	66 443 → 38750 [ACK] Seq=1387 Ack=843 Win=16640 Len=0 TSval=294718697 TSecr=2068906354
18	18.423088164	172.217.28.234	172.29.54.230	TLSv1.3	1091 Application Data, Application Data
19	18.423149157	172.29.54.230	172.217.28.234	TCP	66 38750 → 443 [ACK] Seq=843 Ack=2412 Win=64128 Len=0 TSval=2068906566 TSecr=294718718
20	18.423203751	172.217.28.234	172.29.54.230	TLSv1.3	251 Application Data
21	18.423223930	172.29.54.230	172.217.28.234	TCP	66 38750 → 443 [ACK] Seq=843 Ack=2597 Win=64128 Len=0 TSval=2068906566 TSecr=294718718
22	23.428935903	172.29.54.230	172.217.28.234	TCP	66 38750 → 443 [FIN, ACK] Seq=843 Ack=2597 Win=64128 Len=0 TSval=2068911571 TSecr=294718718
23	23.430329704	172.217.28.234	172.29.54.230	TCP	66 443 → 38750 [FIN, ACK] Seq=2597 Ack=844 Win=16640 Len=0 TSval=294719219 TSecr=2068911571
24	23.430398829	172.29.54.230	172.217.28.234	TCP	66 38750 → 443 [ACK] Seq=844 Ack=2598 Win=64128 Len=0 TSval=2068911573 TSecr=294719219

Figure 3. Captura dos pacotes enviados e recebidos pelo programa. ([Link para melhor visualização](#))

Inicialmente, é feita a busca pelo DNS do site alvo (no caso, www.googleapis.com), e em seguida pode ser observada a troca de pacotes entre o

cliente e o servidor. As mensagens não podem ser visualizadas diretamente no Wireshark por conta da criptografia utilizada na comunicação.



Wireshark · Packet 4 · teste.pcapng

Answers

- > www.googleapis.com: type CNAME, class IN, cname googleapis.l.google.com
- > googleapis.l.google.com: type A, class IN, addr 172.217.28.234
- > googleapis.l.google.com: type A, class IN, addr 172.217.29.10
- > googleapis.l.google.com: type A, class IN, addr 172.217.162.106
- > googleapis.l.google.com: type A, class IN, addr 172.217.28.10
- > googleapis.l.google.com: type A, class IN, addr 172.217.172.138
- > googleapis.l.google.com: type A, class IN, addr 172.217.172.202
- > googleapis.l.google.com: type A, class IN, addr 172.217.28.74
- > googleapis.l.google.com: type A, class IN, addr 216.58.202.10
- > googleapis.l.google.com: type A, class IN, addr 172.217.28.138
- > googleapis.l.google.com: type A, class IN, addr 216.58.202.170
- > googleapis.l.google.com: type A, class IN, addr 172.217.30.74
- > googleapis.l.google.com: type A, class IN, addr 172.217.30.106
- > googleapis.l.google.com: type A, class IN, addr 172.217.29.138
- > googleapis.l.google.com: type A, class IN, addr 172.217.29.170
- > googleapis.l.google.com: type A, class IN, addr 172.217.30.42

Figure 4. Endereços IP adquiridos através do DNS

Segue o link para o vídeo que mostra a execução do programa: https://youtu.be/-hx5z_6OtUE