

Discussion

Program Description

This is a command line program that replicates Curl. It takes a URL as an argument and sends a GET request to the server. It then prints the response header and body. The program uses the socket library.

How to run the program

Run the program by typing `python3 afamjadMyCurl.py [-h] <URL> <hostname (optional)>`.

```
Curl a URL

positional arguments:
  full_URL      http://hostname[ip]:[port]/[query]
  hostname      Optional hostname argument

optional arguments:
  -h, --help    show this help message and exit
```

The program takes in a URL and an optional hostname and issues a GET request to the URL. The program will print whether the GET request was successful or not, output the response body to an HTML file, and log the response (HTTPoutput.HTML and Log.csv).

Note that HTTPS, chunk encoding, and redirection are not supported.

Tests

Test 1: `http://neverssl.com/`

Program output:

```
mininet@mininet-vm:~$ date
Wed Mar 2 13:22:29 PST 2022
mininet@mininet-vm:~$
```

No.	Time	Source	Destination	Protocol	Length	Info
5	0.066811000	10.0.2.15	13.226.219.17	TCP	76	59973 > http [SYN] Seq=0 Win=29200 Len=0
6	0.089372000	13.226.219.17	10.0.2.15	TCP	62	http > 59973 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0
7	0.089893000	10.0.2.15	13.226.219.17	TCP	56	59973 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
8	0.090217000	10.0.2.15	13.226.219.17	HTTP	94	GET / HTTP/1.1
9	0.090854000	13.226.219.17	10.0.2.15	TCP	62	http > 59973 [ACK] Seq=1 Ack=39 Win=6 Len=0
10	0.307724000	13.226.219.17	10.0.2.15	TCP	2976	[TCP segment of a reassembled PDU]
11	0.307784000	10.0.2.15	13.226.219.17	TCP	56	59973 > http [ACK] Seq=39 Ack=2921 Win=0 Len=0
12	0.308081000	13.226.219.17	10.0.2.15	HTTP	1621	HTTP/1.1 200 OK (text/html)
13	0.308131000	10.0.2.15	13.226.219.17	TCP	56	59973 > http [ACK] Seq=39 Ack=4486 Win=0 Len=0
14	0.311212000	10.0.2.15	13.226.219.17	TCP	56	59973 > http [FIN, ACK] Seq=39 Ack=4486 Win=0 Len=0
15	0.311573000	13.226.219.17	10.0.2.15	TCP	62	http > 59973 [ACK] Seq=4486 Ack=40 Win=0 Len=0
16	0.334981000	13.226.219.17	10.0.2.15	TCP	62	http > 59973 [FIN, ACK] Seq=4486 Ack=40 Win=0 Len=0
17	0.335044000	10.0.2.15	13.226.219.17	TCP	56	59973 > http [ACK] Seq=40 Ack=4487 Win=0 Len=0

```
8 0.090217000 10.0.2.15 13.226.219.17 HTTP 94 GET / HTTP/1.1
```

Frame 8: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0
Linux cooked capture
Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 13.226.219.17 (13.226.219.17)
Transmission Control Protocol, Src Port: 59973 (59973), Dst Port: http (80), Seq: 1, Ack: 39, Len: 94
Hypertext Transfer Protocol
GET / HTTP/1.1
[Expert Info (Chat/Sequence): GET / HTTP/1.1]
[Message: GET / HTTP/1.1]
[Severity level: Chat]
[Group: Sequence]
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
Host: neverssl.com

12 0.308081000 13.226.219.17 10.0.2.15 HTTP 1621 HTTP/1.1 200 OK (text/html)

[2 Reassembled TCP Segments (4485 bytes): #10(2920), #12(1565)]
Hypertext Transfer Protocol
HTTP/1.1 200 OK
[Expert Info (Chat/Sequence): HTTP/1.1 200 OK]
[Message: HTTP/1.1 200 OK]
[Severity level: Chat]
[Group: Sequence]
Request Version: HTTP/1.1
Status Code: 200
Response Phrase: OK
Content-Type: text/html
Content-Length: 3961
Connection: keep-alive
Date: Wed, 02 Mar 2022 08:28:46 GMT
Last-Modified: Thu, 04 Nov 2021 18:27:33 GMT
ETag: "41f0211ff315bdd7e2b6dc2e98143935"
Accept-Ranges: bytes
Server: AmazonS3
Vary: Accept-Encoding
Cache-Control: public, max-age=86400
X-Cache: Hit from cloudfront
Via: 1.1 297fb3da326382a83610b8eb79e2222e.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: LAX50-C1
X-Amz-Cf-Id: Kzc30YrEdh6119rqIFFT0s790byQv-gLVpf0QwMqvV040uPiaIqqFg==
Age: 46171
[HTTP response 1/1]
[Time since request: 0.217864000 seconds]
Request in frame: 81
Line-based text data: text/html

Curl output:

The terminal window shows the following commands and output:

```
mininet@mininet-vm:~$ date
Wed Mar 2 13:24:53 PST 2022
mininet@mininet-vm:~$ curl -s http://softwareqatest.com/
# parse the args
urlDict = parseUserInput(args)
to = **
```

The Wireshark packet capture shows the following details:

- Packet 14: GET / HTTP/1.1 (text/html)
- Packet 15: HTTP/1.1 200 OK (text/html)
- Packet 16: GET / HTTP/1.1 (text/html)
- Packet 17: HTTP/1.1 200 OK (text/html)

The packet details for the GET requests show the following headers:

```
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
User-Agent: curl/7.35.0
Host: neverssl.com
Accept: */*
[Full request URI: http://neverssl.com/]
[HTTP request 1/1]
[Response in frame: 14]
```

The packet details for the 200 OK responses show the following headers:

```
Request Version: HTTP/1.1
Status Code: 200
Response Phrase: OK
Content-Type: text/html
Content-Length: 3961
Connection: keep-alive
Date: Wed, 02 Mar 2022 08:28:46 GMT
Last-Modified: Thu, 04 Nov 2021 18:27:33 GMT
ETag: "41f0211ff315bbd7e2b6dc2e98143935"
Accept-Ranges: bytes
Server: AmazonS3
Vary: Accept-Encoding
Cache-Control: public, max-age=86400
X-Cache: Hit from cloudfront
Via: 1.1 29d55562c8b8771afbb951b9f12a386.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: LAX50-C1
X-Amz-Cf-Id: E-HN4UdHIZvLXtalIqgS_GsYk7kMmL0hL4Y10i0HeC88pTqEYFsg==
Age: 46761
```

The program output seems to match the CURL output. the TCP connection was successful, and the response was 200 OK for both commands.

Test 2: <http://softwareqatest.com/>

Program output:

The terminal window shows the following commands and output:

```
mininet@mininet-vm:~$ date
Wed Mar 2 13:30:53 PST 2022
mininet@mininet-vm:~$ curl -s http://softwareqatest.com/
# parse the args
urlDict = parseUserInput(args)
to = **
```

The Wireshark packet capture shows the following details:

- Packet 16: GET / HTTP/1.1 (text/html)
- Packet 17: HTTP/1.1 200 OK (text/html)

The packet details for the GET request show the following headers:

```
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
Host: softwareqatest.com
Accept: */*
[Full request URI: http://softwareqatest.com/]
[HTTP request 1/1]
[Response in frame: 16]
```

The packet details for the 200 OK response show the following headers:

```
Request Version: HTTP/1.1
Status Code: 200
Response Phrase: OK
Date: Wed, 02 Mar 2022 21:30:17 GMT
Server: Apache
Last-Modified: Fri, 05 Feb 2021 03:42:22 GMT
Accept-Ranges: bytes
Content-Length: 10445
Content-Type: text/html
[HTTP response 1/1]
[Time since request: 0.094572000 seconds]
[Request in frame: 8]
```

Curl output:

```
mininet@mininet-vm:~$ date
Wed Mar 2 13:32:57 PST 2022
mininet@mininet-vm:~$
```

No.	Time	Source	Destination	Protocol	Length	Info
3	0.028405000	10.0.2.15	216.92.49.183	TCP	76	46438 > http [SYN] Seq=0 Win=29200 Len=0
4	0.109899000	216.92.49.183	10.0.2.15	TCP	62	http > 46438 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0
5	0.109192000	10.0.2.15	216.92.49.183	TCP	56	46438 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
6	0.109402000	10.0.2.15	216.92.49.183	HTTP	138	GET / HTTP/1.1
7	0.110469000	216.92.49.183	10.0.2.15	TCP	62	http > 46438 [ACK] Seq=1 Ack=83 Win=6 Len=0
8	0.104057000	216.92.49.183	10.0.2.15	TCP	1516	[TCP segment of a reassembled PDU]
9	0.194140000	10.0.2.15	216.92.49.183	TCP	56	46438 > http [ACK] Seq=83 Ack=1461 Win=0 Len=0
10	0.195458000	216.92.49.183	10.0.2.15	TCP	1516	[TCP segment of a reassembled PDU]
11	0.195510000	10.0.2.15	216.92.49.183	TCP	56	46438 > http [ACK] Seq=83 Ack=2921 Win=0 Len=0
12	0.197090000	216.92.49.183	10.0.2.15	TCP	1516	[TCP segment of a reassembled PDU]
13	0.197160000	10.0.2.15	216.92.49.183	TCP	56	46438 > http [ACK] Seq=83 Ack=4381 Win=0 Len=0
14	0.198742000	216.92.49.183	10.0.2.15	TCP	2976	[TCP segment of a reassembled PDU]
15	0.198792000	10.0.2.15	216.92.49.183	TCP	56	46438 > http [ACK] Seq=83 Ack=7301 Win=0 Len=0
16	0.198881000	216.92.49.183	10.0.2.15	HTTP	3389	HTTP/1.1 200 OK (text/html)
17	0.198930000	10.0.2.15	216.92.49.183	TCP	56	46438 > http [ACK] Seq=83 Ack=10634 Win=0 Len=0
18	0.199615000	10.0.2.15	216.92.49.183	TCP	56	46438 > http [FIN, ACK] Seq=83 Ack=10634 Win=0 Len=0
19	0.201822000	216.92.49.183	10.0.2.15	TCP	62	http > 46438 [ACK] Seq=10634 Ack=84 Win=0 Len=0
20	0.290385000	216.92.49.183	10.0.2.15	TCP	62	http > 46438 [FIN, ACK] Seq=10634 Ack=84 Win=0 Len=0
21	0.290449000	10.0.2.15	216.92.49.183	TCP	56	46438 > http [ACK] Seq=84 Ack=10635 Win=0 Len=0

```
> Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
User-Agent: curl/7.35.0\r\n
Host: softwareqatest.com\r\n
Accept: */*\r\n
\r\n
[Full request URI: http://softwareqatest.com/]
[HTTP request 1/1]
[Response in frame: 16]
16 0.198881000 216.92.49.183 10.0.2.15 HTTP 3389 HTTP/1.1 200 OK (text/html)
> Internet Protocol Version 4, Src: 216.92.49.183 (216.92.49.183), Dst: 10.0.2.15 (10.0.2.15)
> Transmission Control Protocol, Src Port: http (80), Dst Port: 46438 (46438), Seq: 7301, Len: 3389
> 5 Reassembled TCP Segments (10633 bytes): #8(1460), #10(1460), #12(1460), #14(2920), #16(1460)
> Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
[Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
Request Version: HTTP/1.1
Status Code: 200
Response Phrase: OK
Date: Wed, 02 Mar 2022 21:31:42 GMT\r\n
Server: Apache\r\n
Last-Modified: Fri, 05 Feb 2021 03:42:22 GMT\r\n
Accept-Ranges: bytes\r\n
Content-Length: 10445\r\n
Content-Type: text/html\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.089479000 seconds]
[Request in frame: 6]
```

The program output seems to match the CURL output. the TCP connection was successful, and the response was 200 OK for both commands.

Test 3: <http://www.example.com/anyname.html>

Program output:

```
mininet@mininet-vm:~$ date
Wed Mar 2 13:34:29 PST 2022
mininet@mininet-vm:~$
```

No.	Time	Source	Destination	Protocol	Length	Info
5	0.034519000	10.0.2.15	93.184.216.34	TCP	76	36353 > http [SYN] Seq=0 Win=29200 Len=0
6	0.052204000	93.184.216.34	10.0.2.15	TCP	62	http > 36353 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0
7	0.052320000	10.0.2.15	93.184.216.34	TCP	56	36353 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
8	0.052523000	10.0.2.15	93.184.216.34	HTTP	109	GET /anyname.html HTTP/1.1
9	0.052819000	93.184.216.34	10.0.2.15	TCP	62	http > 36353 [ACK] Seq=1 Ack=54 Win=6 Len=0
10	0.104416000	93.184.216.34	10.0.2.15	HTTP	1653	HTTP/1.1 404 Not Found (text/html)
11	0.104434000	10.0.2.15	93.184.216.34	TCP	56	36353 > http [ACK] Seq=54 Ack=1598 Win=0 Len=0
12	0.107147000	10.0.2.15	93.184.216.34	TCP	56	36353 > http [RST, ACK] Seq=54 Ack=1598 Win=0 Len=0

```
> Frame 8: 109 bytes on wire (872 bits), 109 bytes captured (872 bits) on interface 0
> Linux cooked capture
> Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 93.184.216.34 (93.184.216.34)
> Transmission Control Protocol, Src Port: 36353 (36353), Dst Port: http (80), Seq: 1, Ack: 1598, Len: 109
> Hypertext Transfer Protocol
GET /anyname.html HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET /anyname.html HTTP/1.1\r\n]
Request Method: GET
Request URI: /anyname.html
Request Version: HTTP/1.1
Host: www.example.com\r\n
\r\n
[Full request URI: http://www.example.com/anyname.html]
[HTTP request 1/1]
[Response in frame: 10]
10 0.052523000 10.0.2.15 93.184.216.34 HTTP 109 GET /anyname.html HTTP/1.1
> Transmission Control Protocol, Src Port: http (80), Dst Port: 36353 (36353), Seq: 1, Ack: 1598, Len: 109
> Hypertext Transfer Protocol
HTTP/1.1 404 Not Found\r\n
[Expert Info (Chat/Sequence): HTTP/1.1 404 Not Found\r\n]
Request Version: HTTP/1.1
Status Code: 404
Response Phrase: Not Found
Accept-Ranges: bytes\r\n
Age: 93184\r\n
Cache-Control: max-age=604800\r\n
Content-Type: text/html; charset=UTF-8\r\n
Date: Wed, 02 Mar 2022 21:33:45 GMT\r\n
Expires: Wed, 09 Mar 2022 21:33:45 GMT\r\n
Last-Modified: Tue, 01 Mar 2022 19:42:01 GMT\r\n
Server: ECS (sab/5718)\r\n
Vary: Accept-Encoding\r\n
X-Cache: 404-HIT\r\n
Content-Length: 1256\r\n
\r\n
[HTTP response 1/1]
[Time since request: 0.051893000 seconds]
[Request in frame: 8]
Line-based text data: text/html
0000 00 00 00 01 00 05 52 54 00 12 35 02 00 00 08 00 .....RT .S....
0010 45 00 06 65 00 9b 00 00 40 06 32 0f 5d b8 d8 22 E..e...@.2.1...
0020 0a 00 02 0f 00 50 8e 01 00 b7 98 02 76 e7 4d 1b ....P...v.M.
0030 50 18 ff ff 48 41 00 00 48 54 54 50 2f 31 2e 31 P...HA.. HTTP/1.1
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

Curl output:

The screenshot displays two windows. The top window is a terminal showing the output of the 'date' command and the 'curl' command. The bottom window is Wireshark, showing a packet capture of an HTTP GET request to `/anyname.html` and a 404 Not Found response. The response details show the status code 404 and the phrase 'Not Found'. The packet list shows the request and response packets.

```
mininet@mininet-vm:~$ date
Wed Mar 2 13:39:42 PST 2022
mininet@mininet-vm:~$ curl -v http://www.example.com/anyname.html
* Host www.example.com: not found
* Connected to www.example.com (10.0.2.15) port 80
> GET /anyname.html HTTP/1.1
> Host: www.example.com
> User-Agent: curl/7.35.0
> Accept: */*
* HTTP request 1/1
* HTTP response 1/1
[Response in frame: 10]
```

Wireshark packet details:

- Frame 6: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits) on interface 0
- Linux cooked capture
- Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 93.184.216.34 (93.184.216.34)
- Transmission Control Protocol, Src Port: 36354 (36354), Dst Port: http (80), Seq: 1, Ack: 1
- Hypertext Transfer Protocol
 - GET /anyname.html HTTP/1.1
 - [Expert Info (Chat/Sequence): GET /anyname.html HTTP/1.1]
 - Request Method: GET
 - Request URI: /anyname.html
 - Request Version: HTTP/1.1
 - User-Agent: curl/7.35.0
 - Host: www.example.com
 - Accept: */*

Wireshark packet details (Response):

- Frame 10: 193 bytes on wire (1544 bits), 193 bytes captured (1544 bits) on interface 0
- Linux cooked capture
- Internet Protocol Version 4, Src: 93.184.216.34 (93.184.216.34), Dst: 10.0.2.15 (10.0.2.15)
- Transmission Control Protocol, Src Port: http (80), Dst Port: 36354 (36354), Seq: 92, Ack: 1599, Win: 0, Len: 0
- Hypertext Transfer Protocol
 - HTTP/1.1 404 Not Found
 - [Expert Info (Chat/Sequence): HTTP/1.1 404 Not Found]
 - Request Version: HTTP/1.1
 - Status Code: 404
 - Response Phrase: Not Found
 - Accept-Ranges: bytes
 - Age: 93250
 - Cache-Control: max-age=604800
 - Content-Type: text/html; charset=UTF-8
 - Date: Wed, 02 Mar 2022 21:36:11 GMT
 - Expires: Wed, 09 Mar 2022 21:36:11 GMT
 - Last-Modified: Tue, 01 Mar 2022 19:42:01 GMT
 - Server: ECS (sab/5718)
 - Vary: Accept-Encoding
 - X-Cache: 404-HIT
 - Content-Length: 1256

Both our program and Curl establish a TCP connection, send the GET request, and receive a response of 404 Not Found. However, my program closes the socket upon an unsuccessful response, so that is why there is an RST packet. An RST packet indicates that the client no longer needs data from the server. Curl, on the other hand, continues to receive content from the server.

Test 4: `http://www.example.com/index.html`

Program output:

The screenshot displays two windows. The top window is a terminal showing the output of the 'date' command and the 'curl' command. The bottom window is Wireshark, showing a packet capture of an HTTP GET request to `/index.html` and a 403 Forbidden response. The response details show the status code 403 and the phrase 'Forbidden'. The packet list shows the request and response packets.

```
mininet@mininet-vm:~$ date
Wed Mar 2 13:41:40 PST 2022
mininet@mininet-vm:~$ curl -v http://www.example.com/index.html
* Host www.example.com: not found
* Connected to www.example.com (10.0.2.15) port 80
> GET /index.html HTTP/1.1
> Host: www.example.com
> User-Agent: curl/7.35.0
> Accept: */*
* HTTP request 1/1
* HTTP response 1/1
[Response in frame: 10]
```

Wireshark packet details:

- Frame 8: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
- Linux cooked capture
- Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 104.247.82.13 (104.247.82.13)
- Transmission Control Protocol, Src Port: 59357 (59357), Dst Port: http (80), Seq: 1, Ack: 1
- Hypertext Transfer Protocol
 - GET /index.html HTTP/1.1
 - [Expert Info (Chat/Sequence): GET /index.html HTTP/1.1]
 - Request Method: GET
 - Request URI: /index.html
 - Request Version: HTTP/1.1
 - Host: www.example.com

Wireshark packet details (Response):

- Frame 10: 350 bytes on wire (2800 bits), 350 bytes captured (2800 bits) on interface 0
- Linux cooked capture
- Internet Protocol Version 4, Src: 104.247.82.13 (104.247.82.13), Dst: 10.0.2.15 (10.0.2.15)
- Transmission Control Protocol, Src Port: http (80), Dst Port: 59357 (59357), Seq: 1, Ack: 1
- Hypertext Transfer Protocol
 - HTTP/1.1 403 Forbidden
 - [Expert Info (Chat/Sequence): HTTP/1.1 403 Forbidden]
 - Request Version: HTTP/1.1
 - Status Code: 403
 - Response Phrase: Forbidden
 - Server: nginx
 - Date: Wed, 02 Mar 2022 21:36:51 GMT
 - Content-Type: text/html
 - Content-Length: 146
 - Connection: keep-alive

Curl output:

The image shows a Wireshark packet capture of an HTTP GET request to `/index.html` on `104.247.82.13` from `10.0.2.15`. The response is a `403 Forbidden` status code. The left pane shows the packet list and packet details, while the right pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.125094000	10.0.2.15	104.247.82.13	TCP	76	59358 > http [SYN] Seq=0 Win=29200 Len=0
4	0.207791000	104.247.82.13	10.0.2.15	TCP	62	http > 59358 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0
5	0.207871000	10.0.2.15	104.247.82.13	TCP	56	59358 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
6	0.208052000	10.0.2.15	104.247.82.13	HTTP	144	GET /index.html HTTP/1.1
7	0.209123000	104.247.82.13	10.0.2.15	TCP	62	http > 59358 [ACK] Seq=1 Ack=89 Win=0 Len=0
8	0.376323000	104.247.82.13	10.0.2.15	HTTP	350	HTTP/1.1 403 Forbidden (text/html)

Packet 8 Details:

- Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 104.247.82.13 (104.247.82.13)
- Transmission Control Protocol, Src Port: 59358 (59358), Dst Port: http (80), Seq: 1, Ack: 1
- Hypertext Transfer Protocol
 - GET /index.html HTTP/1.1\r\n
 - [Expert Info (Chat/Sequence): GET /index.html HTTP/1.1\r\n]
 - Request Method: GET
 - Request URI: /index.html
 - Request Version: HTTP/1.1
 - User-Agent: curl/7.35.0\r\n
 - Host: www.example.com\r\n
 - Accept: */*\r\n
 - \r\n
 - [Full request URI: http://www.example.com/index.html]
 - [HTTP request 1/1]
 - [Response in frame: 8]

Packet 8 Raw Data:

```
8 0.376323000 104.247.82.13 10.0.2.15 HTTP 350 HTTP/1.1 403 Forbidden (text/html)
```

This output comparison is similar to test 3, except that instead of returning a 404 Not Found, the program and Curl return 403 Forbidden.

Test 5: <http://www.nintendo.com>

Program output:

The image shows a Wireshark packet capture of an HTTP GET request to `/` on `151.101.42.132` from `10.0.2.15`. The response is a `301 Moved Permanently` status code. The left pane shows the packet list and packet details, while the right pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
5	0.034096000	10.0.2.15	151.101.42.132	TCP	76	34110 > http [SYN] Seq=0 Win=29200 Len=0
6	0.050158000	151.101.42.132	10.0.2.15	TCP	62	http > 34110 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0
7	0.050279000	10.0.2.15	151.101.42.132	TCP	56	34110 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
8	0.050347000	10.0.2.15	151.101.42.132	HTTP	98	GET / HTTP/1.1
9	0.050578000	151.101.42.132	10.0.2.15	TCP	62	http > 34110 [ACK] Seq=1 Ack=43 Win=0 Len=0
10	0.098477000	151.101.42.132	10.0.2.15	HTTP	421	HTTP/1.1 301 Moved Permanently

Packet 10 Details:

- Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 151.101.42.132 (151.101.42.132)
- Transmission Control Protocol, Src Port: 34110 (34110), Dst Port: http (80), Seq: 1, Ack: 1
- Hypertext Transfer Protocol
 - GET / HTTP/1.1\r\n
 - [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
 - Request Method: GET
 - Request URI: /
 - Request Version: HTTP/1.1
 - Host: www.nintendo.com\r\n
 - \r\n
 - [Full request URI: http://www.nintendo.com/]
 - [HTTP request 1/1]
 - [Response in frame: 10]

Packet 10 Raw Data:

```
10 0.098477000 151.101.42.132 10.0.2.15 HTTP 421 HTTP/1.1 301 Moved Permanently
```


Curl output:

mininet@mininet-vm:~\$ date
Wed Mar 2 13:47:25 PST 2022
mininet@mininet-vm:~\$

No.	Time	Source	Destination	Protocol	Length	Info
3	0.028407000	10.0.2.15	151.101.190.132	TCP	76	54091 > http [SYN] Seq=0 Win=29200 Len=0
4	0.043424000	151.101.190.132	10.0.2.15	TCP	62	http > 54091 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0
5	0.043519000	10.0.2.15	151.101.190.132	TCP	56	54091 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
6	0.043690000	10.0.2.15	151.101.190.132	HTTP	136	GET / HTTP/1.1
7	0.043974000	151.101.190.132	10.0.2.15	TCP	62	http > 54091 [ACK] Seq=1 Ack=81 Win=65536 Len=0
8	0.061475000	151.101.190.132	10.0.2.15	HTTP	421	HTTP/1.1 301 Moved Permanently
9	0.061533000	10.0.2.15	151.101.190.132	TCP	56	54091 > http [ACK] Seq=81 Ack=366 Win=65536 Len=0
10	0.061737000	151.101.190.132	10.0.2.15	TCP	62	http > 54091 [FIN, ACK] Seq=366 Ack=81 Win=0 Len=0
11	0.061981000	10.0.2.15	151.101.190.132	TCP	56	54091 > http [FIN, ACK] Seq=81 Ack=366 Win=0 Len=0
12	0.062281000	151.101.190.132	10.0.2.15	TCP	62	http > 54091 [ACK] Seq=367 Ack=82 Win=65536 Len=0

6 0.043690000 10.0.2.15 151.101.190.132 HTTP 136 GET / HTTP/1.1

8 0.061475000 151.101.190.132 10.0.2.15 HTTP 421 HTTP/1.1 301 Moved Permanently

Frame 6: 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits) on interface 0
Linux cooked capture
Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 151.101.190.132 (151.101.132)
Transmission Control Protocol, Src Port: 54091 (54091), Dst Port: http (80), Seq: 1, Ack: 81, Win: 29200, Len: 0
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
User-Agent: curl/7.35.0\r\nHost: www.nintendo.com\r\nAccept: */*\r\n\r\n[Full request URI: http://www.nintendo.com/]
[HTTP request 1/1]
[Response in frame: 8]

Frame 8: 421 bytes on wire (3368 bits), 421 bytes captured (3368 bits) on interface 0
Linux cooked capture
Internet Protocol Version 4, Src: 151.101.190.132 (151.101.190.132), Dst: 10.0.2.15 (10.0.2.15)
Transmission Control Protocol, Src Port: http (80), Dst Port: 54091 (54091), Seq: 1, Ack: 82, Win: 65536, Len: 421
Hypertext Transfer Protocol
HTTP/1.1 301 Moved Permanently\r\n
[Expert Info (Chat/Sequence): HTTP/1.1 301 Moved Permanently\r\n]
Request Version: HTTP/1.1
Status Code: 301
Response Phrase: Moved Permanently
Server: Varnish\r\nRetry-After: 0\r\nLocation: https://www.nintendo.com/\r\nContent-Length: 0\r\nAccept-Ranges: bytes\r\nDate: Wed, 02 Mar 2022 21:46:19 GMT\r\nVia: 1.1 varnish\r\nConnection: close\r\nX-Served-By: cache-pao17428-PAO\r\nX-Cache: HIT\r\nX-Cache-Hits: 0\r\nX-Timer: S1646257580.788865,VS0,VE0

The program and Curl output match. An HTTP GET request is successfully sent, and a response of 301 Moved Permanently is returned. The TCP connection is established and closed.

Test 6: http://www.google.com:443

Program output:

mininet@mininet-vm:~\$ date
Wed Mar 2 13:52:10 PST 2022
mininet@mininet-vm:~\$

No.	Time	Source	Destination	Protocol	Length	Info
5	0.036947000	10.0.2.15	142.251.46.228	TCP	76	54257 > https [SYN] Seq=0 Win=29200 Len=0
6	0.051406000	142.251.46.228	10.0.2.15	TCP	62	https > 54257 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0
7	0.051481000	10.0.2.15	142.251.46.228	TCP	56	54257 > https [ACK] Seq=1 Ack=1 Win=29200 Len=0
8	0.051686000	10.0.2.15	142.251.46.228	SSL	96	Continuation Data
9	0.051959000	142.251.46.228	10.0.2.15	TCP	62	https > 54257 [ACK] Seq=1 Ack=41 Win=65536 Len=0
10	0.068001000	142.251.46.228	10.0.2.15	TCP	62	https > 54257 [FIN, ACK] Seq=1 Ack=41 Win=0 Len=0
11	0.068939000	10.0.2.15	142.251.46.228	TCP	56	54257 > https [FIN, ACK] Seq=41 Ack=2 Win=0 Len=0
12	0.069254000	142.251.46.228	10.0.2.15	TCP	62	https > 54257 [ACK] Seq=2 Ack=42 Win=65536 Len=0

Curl output:

mininet@mininet-vm:~\$ date
Wed Mar 2 13:52:49 PST 2022
mininet@mininet-vm:~\$

No.	Time	Source	Destination	Protocol	Length	Info
3	0.028537000	10.0.2.15	142.251.46.228	TCP	76	54258 > https [SYN] Seq=0 Win=29200 Len=0
4	0.044814000	142.251.46.228	10.0.2.15	TCP	62	https > 54258 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0
5	0.044885000	10.0.2.15	142.251.46.228	TCP	56	54258 > https [ACK] Seq=1 Ack=1 Win=29200 Len=0
6	0.045044000	10.0.2.15	142.251.46.228	SSL	138	Continuation Data
7	0.046078000	142.251.46.228	10.0.2.15	TCP	62	https > 54258 [ACK] Seq=1 Ack=83 Win=65536 Len=0
8	0.067415000	142.251.46.228	10.0.2.15	TCP	62	https > 54258 [FIN, ACK] Seq=1 Ack=83 Win=0 Len=0
9	0.067637000	10.0.2.15	142.251.46.228	TCP	56	54258 > https [FIN, ACK] Seq=83 Ack=2 Win=0 Len=0
10	0.067930000	142.251.46.228	10.0.2.15	TCP	62	https > 54258 [ACK] Seq=2 Ack=84 Win=65536 Len=0

The program output seems to match the CURL output. Wireshark doesn't seem to be outputting HTML packets for either program. What we can see from these outputs, however, is that there is an SSL packet, which indicates that port 443 was used.