

Inlämning 1

Olof Jönsson, oljn22

19901115-0712

ET1447

2025-02-06

Uppgift A4

Jag hade inte tillgång till två vanliga datorer i hemmet, så jag funderade på att köra med en annan student över internet. Jag läste på lite, och det verkade krävas ändringar i router/brandvägg, och där kände jag mig osäker på om jag skulle röra till något, eller öppna upp något säkerhetshål.

Jag löste det sedan med att installera en python editor på min ipad. Jag lät klienten vara på ipaden och servern på datorn. Först fick jag det inte att fungera, och jag testa att skapa regler i brandväggen. Det löste sig sedan genom att jag ändrade mitt hemma-wifi till privat nätverk, och då kunde jag plocka bort reglerna i brandväggen.

Jag undersökte även trafiken i Wireshark, och kunde konstatera att TCP-varianten gav upphov till tolv rader jämfört med UDP med två.

```

#!/usr/bin/env python3

from socket import *

class CustomWebBrowser:
    """ Simple custom web browser
        Author: Olof Jönsson, oljn22 """

    def __init__(self):
        """ Set name, and create TCP socket """
        self._name = "CustomWebBrowser/1.0"
        self._socket = socket(AF_INET, SOCK_STREAM)

    def _connect(self, host, port=80):
        """ Connect to server host """
        self._socket.connect((host, port))

    def _send(self, header):
        """ Send encoded header into socket """
        print("<<< SENDING REQUEST >>>\n")
        print(header)
        self._socket.send(header.encode())

    def _receive_and_print(self):
        """ Receive response from server host, decode and print """
        response = self._socket.recv(1024)
        print("<<< RECEIVING RESPONSE >>>\n")
        print(response.decode())

    def _close(self):
        """ Close socket """
        self._socket.close()

    def _build_header(
        self,
        host,
        path,
        query,
        method="GET",
        protocol="HTTP/1.1",
        accept="text/html",
        conn="close"
    ):
        """ Build a simple HTTP header """
        header = f"{method} {path}{query} {protocol}\r\n"
        header += f"Host: {host}\r\n"
        header += f"User-Agent: {self._name}\r\n"
        header += f"Accept: {accept}\r\n"
        header += f"Connection: {conn}\r\n"
        header += "\r\n"

        return header

    def request_page(self, host, path, query):

```

```
""" Request a web page """
self._connect(host)
self._send(self._build_header(host, path, query))
self._receive_and_print()
self._close()
```

```
if __name__ == "__main__":
    # Resource to request
    host = "www.ingonline.nu"
    path = "/tictactoe/index.php"
    query = "?board=xoxoxoeeee"

    # Init browser, and request page
    browser = CustomWebBrowser()
    browser.request_page(host, path, query)
```

Tid	192.168.1.223	46.30.213.174	Kommentar
4.030281	56766	56766 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_...	TCP: 56766 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 ..
4.040837	56766	80 → 56766 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=...	TCP: 80 → 56766 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0..
4.041033	56766	56766 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=273...	TCP: 56766 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval.
4.041170	56766	GET /tictactoe/index.php?board=xoxoxoeee HTTP/1.1	HTTP: GET /tictactoe/index.php?board=xoxoxoeee HTTP/1.1
4.052670	56766	80 → 56766 [ACK] Seq=1 Ack=150 Win=65024 Len=0 TSval=3...	TCP: 80 → 56766 [ACK] Seq=1 Ack=150 Win=65024 Len=0 TS.
4.101799	56766	HTTP/1.1 200 OK (text/html)	HTTP: HTTP/1.1 200 OK (text/html)
4.102010	56766	56766 → 80 [ACK] Seq=150 Ack=539 Win=64128 Len=0 TSval...	TCP: 56766 → 80 [ACK] Seq=150 Ack=539 Win=64128 Len=0 .
4.102200	56766	56766 → 80 [FIN, ACK] Seq=150 Ack=539 Win=64128 Len=0 T...	TCP: 56766 → 80 [FIN, ACK] Seq=150 Ack=539 Win=64128 Le.
4.105659	56766	80 → 56766 [FIN, ACK] Seq=539 Ack=150 Win=65024 Len=0 T...	TCP: 80 → 56766 [FIN, ACK] Seq=539 Ack=150 Win=65024 Le.
4.105954	56766	56766 → 80 [ACK] Seq=151 Ack=540 Win=64128 Len=0 TSval...	TCP: 56766 → 80 [ACK] Seq=151 Ack=540 Win=64128 Len=0 .
4.112697	56766	80 → 56766 [ACK] Seq=540 Ack=151 Win=65024 Len=0 TSval...	TCP: 80 → 56766 [ACK] Seq=540 Ack=151 Win=65024 Len=0 .

> Frame 52: 215 bytes on wire (1720 bits), 215 bytes captured (1720 bits) on interface \Device\NPF_{CF49DF29-C62B-474B-94A0-8F4201885A08}, id 0
> Ethernet II, Src: Intel_5e:00:b0 (18:5e:0f:5e:00:b0), Dst: ASUSTekCOMPU_0d:8f:30 (70:4d:7b:0d:8f:30)
> Internet Protocol Version 4, Src: 192.168.1.223, Dst: 46.30.213.174
> Transmission Control Protocol, Src Port: 56766, Dst Port: 80, Seq: 1, Ack: 1, Len: 149

▼ Hypertext Transfer Protocol

> GET /tictactoe/index.php?board=xoxoxoeee HTTP/1.1\r\n
Host: www.ingonline.nu\r\n
User-Agent: CustomWebBrowser/1.0\r\n
Accept: text/html\r\n
Connection: close\r\n
\r\n

[Response in frame: 54]

[Full request URI: <http://www.ingonline.nu/tictactoe/index.php?board=xoxoxoeee>]

▼ Hypertext Transfer Protocol

> HTTP/1.1 200 OK\r\n
Date: Wed, 05 Feb 2025 15:02:10 GMT\r\n
Server: Apache\r\n
X-Powered-By: PHP/8.0.30\r\n
Vary: Accept-Encoding\r\n
Content-Type: text/html; charset=UTF-8\r\n
X-Onecom-Cluster-Name: \r\n
X-Varnish: 1325731166\r\n
Age: 0\r\n
Via: 1.1 webcache2 (Varnish/trunk)\r\n
Accept-Ranges: bytes\r\n
> Content-Length: 223\r\n
Connection: close\r\n

\r\n

[\[Request in frame: 52\]](#)

[Time since request: 0.060629000 seconds]

[Request URI: /tictactoe/index.php?board=xoxoxoeee]

[\[Full request URI: http://www.ingonline.nu/tictactoe/index.php?board=xoxoxoeee\]](#)

File Data: 223 bytes

▼ Line-based text data: text/html (11 lines)

<html>\r\n
\r\n
<head>\r\n
<TITLE>Tic-Tac-Toe</TITLE>\r\n
</head>\r\n
\r\n
<body>\r\n
\r\n
 Tic-Tac-Toe (3x3)
\r\n

<u>Before turn</u>
x o x
o x o
e e e

<u>After turn</u>
e o x
o x o
x e e
</body>\r\n
</html>

Uppgift B4

Det gick bra att bygga en enkel webbläsare med socketprogrammering i python. Jag lät webbläsaren bli ett objekt (*CustomWebBrowser*), med privata metoder för varje enskild interaktion, och den publika metoden *request_page()*. När objektet instansieras skapas en TCP socket. När *request_page()* anropas med host, path och query, upprättas en anslutning till hosten och en http-request-header byggs (GET, HTTP/1.1, Connection: close). Headern skickas över anslutningen, och sedan tas ett http-response emot och skrivs ut.