

# WebProxyServer UDP Protokoll beskrivelse

Kenneth Fossen @nix007

20. mars 2018

## Innhold

<b>1</b>	<b>Innledning</b>	<b>2</b>
<b>2</b>	<b>Protokoll beskrivelse for kommunikasjon med WPS</b>	<b>3</b>
2.1	Kommunikasjon . . . . .	3
2.1.1	Klienten . . . . .	3
2.2	Vertnavn håndtering og Stihåndtering . . . . .	4
2.3	Feilhåndtering . . . . .	6
2.4	Protokoll beskrivelse . . . . .	6
2.5	Skisse av kommunikasjon . . . . .	7

## Figurer

1	Klient meny . . . . .	3
2	Klient henter kefo.no og kefo.no/ . . . . .	5
3	Klient get request . . . . .	6
4	Klient WPS Kommunikasjon . . . . .	7

## 1 Innledning

Oppgaven beskriver at vi skal lage en WebProxyServer (WPS) der all kommunikasjon mellom klienten (K) og WPS blir håndtert via UDP.

I fra oppgaveteksten: *K skal kommuniserer med WPS gjennom en egen applikasjonsprotokoll som dere skal beskrive som en egen deloppgave. Denne protokollen skal dere implemterer gjennom at K og WPS kommuniserer ved hjelp av UDP/DatagramSocketer.*

Videre står det at: *K skal sende **vertsnavn** og eventuelt **stinavn** til WPS. Beskriv spesielt hvordan protokollen håndterer situasjonen med at det er valgfritt for K å sende stinavn.*

Beskrivelse:

- WPS skal bruke TCP for å hente kun *headerinfo* med RAW HTTP fra en webserver, angitt med bruker input i fra K. K skal også kunne oppgi stinavn som skal hentes på server. Dersom K ikke oppgir stinavn skal default /.
- K skal kommunisere med WPS gjennom egen applikasjonsprotokoll. Gjøres ved hjelp av UDP/DatagramSocketer.

## 2 Protokoll beskrivelse for kommunikasjon med WPS

### 2.1 Kommunikasjon

#### 2.1.1 Klienten

##### Kommandoer

- **GET** - henter en websti
- **?** - viser hjelpemeny
- **QUIT** - avslutter applikasjonen

Klienten er satt opp til å kommunisere med WPS Serveren via port UDP/8080. Denne må være åpnet for i brannmur for at kommunikasjonen skal fungere mellom klient og server.

**GET** : brukes på en av de følgende måte: *GET www.uib.no GET www.uib.no/matnat GET http://www.bt.no GET https://www.bt.no/*

Dette vil returnere headerinfo i fra *websiden* og vise det på klienten. Serveren vil da lage en HEAD post som blir sendt på enten HTTP eller HTTPS til serveren.

Klienten kan så requeste en ny eller avslutte applikasjonen sin med kommandoen **QUIT** eller **Q**.

```
eduroam-nat-135:client spydx$ javac WPSCClient.java ; java WPSCClient
WPSCClient v1.00 nix007 - Ready

Help menu:

GET hostname - request a webpage e.g: GET www.example.com
GET hostname/path - request a subpage e.g GET www.uib.no/matnat/
GET http://hostname - request a webpage e.g GET http://www.kefo.no
GET https://hostname - request a secure webpage e.g GET https://www.kefo.no

? - this menu
Q - end application

[k] [Connected] > q
eduroam-nat-135:client spydx$
```

Figur 1: Klient meny

## 2.2 Vertnavn håndtering og Stihåndtering

Serveren tar i mot en request på følgende format:

«*GET URL*»

«*GET www.kefo.no*» eller «*GET www.kefo.no/wp-admin*»

Når servern mottar denne GET kommandoen blir GET og eventuelt protokoll spesifikasjon HTTP eller HTTPS strippet fra hele kommandoen, slik at en står igjen med URLen. URLen blir ACKet i en pakke som ser slik ut ACK URL. I dette tilfellet «ACK WWW.KEFO.NO» og vises på serverconsolet.

Så blir URL dataene fra kommandoen behandlet i processURL(). Dette gjør at når vi skal sende HTTP HEAD post over TCP så kan vi konstruere den etter HTTP standarden.

I getHttpHeaders() blir det også tatt et protokollvalg, her bestemmes det om det skal åpnes en HTTP eller HTTPS tilkobling til websiden.

Vi følger *HTTP/1.1* standarden. Derfor blir Headeren *Connection: close* lagt til siden vi ikke skal bruke *persistent connections*.

For en helt vanlig sti 1. «www.kefo.no» eller 2. «kefo.no/» så blir det slik brukeren skriver.

- I 1. tilfellet får en få HTTP/1.1 301/Moved Permanently
- I 2. tilfellet vil en få HTTP/1.1 200 OK

Se Figur (2)

WPS Servern vil kunne håndtere følgende situasjoner:

- webside med sti
- websiden uten sti
- HTTP og HTTPS om spesifisert av bruker
- ugyldige web adresser (FQDN)

WPS Serveren begrenser også til 5 tegn på en webadresse e.g ba.no.

Når serveren skal sende HTTP HEAD til en webside uten sti, så blir requesten formet slik: *HEAD / HTTP/1.1*

Når serveren skal sende HTTP HEAD til en webside med sti, så blir requesten formet slik: *HEAD http://FQDN/path HTTP/1.1* og for HTTPS blir den slik: *HEAD https://FQDN/path HTTP/1.1*.

```
eduroam-nat-135:client spydx$ javac WPSCClient.java ; java WPSCClient
WPSCClient v1.00 nix007 - Ready

Help menu:
GET hostname - request a webpage e.g: GET www.example.com
GET hostname/path - request a subpage e.g GET www.uib.no/matnat/
GET http://hostname - request a webpage e.g GET http://www.kefo.no
GET https://hostname - request a secure webpage e.g GET https://www.kefo.no

? - this menu
Q - end application

[k] [Connected] > get www.kefo.no
[k] Sending packet to: /127.0.0.1
[k] Waiting for server ACK
[Server] [ACK, WWW.KEFO.NO]
[k] Waiting for server HEADER
[Server] HEADER dump:

HTTP/1.1 301 Moved Permanently
Server: nginx
Date: Tue, 20 Mar 2018 14:03:17 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.5.36
Location: http://kefo.no/

[k] [Connected] > get kefo.no/
[k] Sending packet to: /127.0.0.1
[k] Waiting for server ACK
[Server] [ACK, KEFO.NO/]
[k] Waiting for server HEADER
[Server] HEADER dump:

HTTP/1.1 200 OK
Server: nginx
Date: Tue, 20 Mar 2018 14:04:08 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.5.36
Link: <http://kefo.no/wp-json/>; rel="https://api.w.org/", <https://wp.me/92qpF>
; rel=shortlink

[k] [Connected] > █
```

Figur 2: Klient henter kefo.no og kefo.no/

## 2.3 Feilhåndtering

Alle feilkoder i fra HTTP protokollen blir returnert som meldinger til brukeren.

Feil, e.g FQDN feil, så vil WPS Serveren sende en RST pakke til klienten og denne vil så avbryte henting og kan gjøre et nytt forsøk.

## 2.4 Protokoll beskrivelse

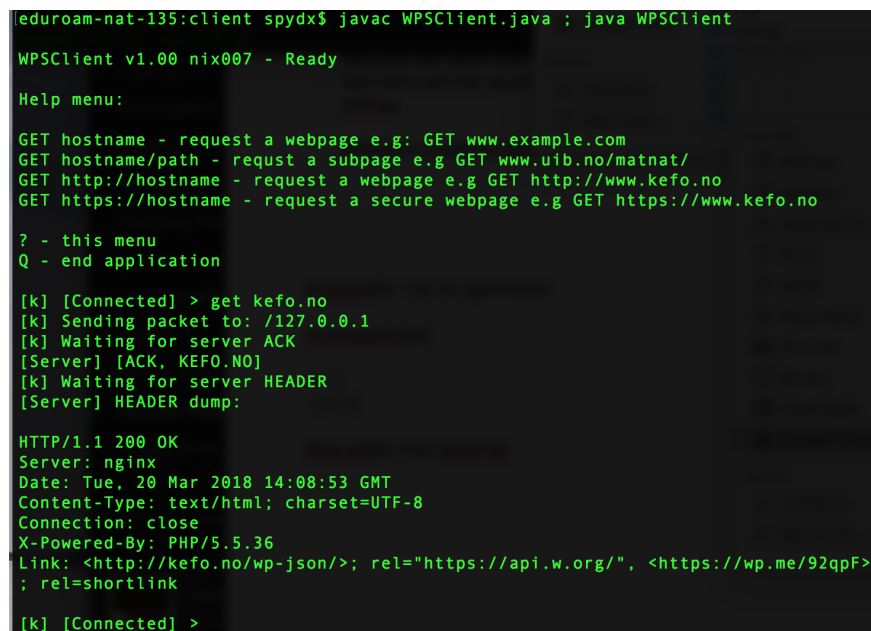
Siden vi kjører UDP mellom klienten, så er det ikke noen avklaring i fra serveren om at den er klar, dette medfører at klienten henger til den får kontakt med serveren. Det er ikke noe Timeout i klienten.

Dette er følgende kommunikasjon som er for en GET request i fra klienten:

- K : get www.webside.no to WPS
- WPS : ack to K
- K : Waiting for headers from WPS
- WPS : TCP request www.webside.no from WEB
- WEB : TCP response to WPS
- WPS : Headerdata to K

Det er likt for HTTP og HTTPS.

Dette vil se slik ut på en klient:



```
eduroam-nat-135:client spydx$ javac WPSCClient.java ; java WPSCClient
WPSCClient v1.00 nix007 - Ready

Help menu:

GET hostname - request a webpage e.g: GET www.example.com
GET hostname/path - request a subpage e.g GET www.uib.no/matnat/
GET http://hostname - request a webpage e.g GET http://www.kefo.no
GET https://hostname - request a secure webpage e.g GET https://www.kefo.no

? - this menu
Q - end application

[k] [Connected] > get kefo.no
[k] Sending packet to: /127.0.0.1
[k] Waiting for server ACK
[Server] [ACK, KEFO.NO]
[k] Waiting for server HEADER
[Server] HEADER dump:

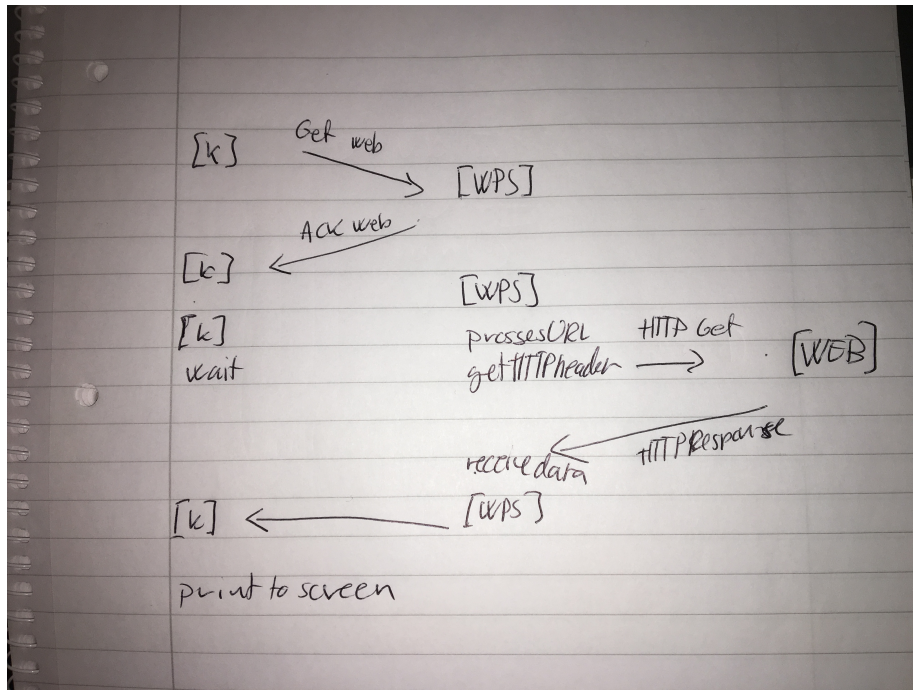
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 20 Mar 2018 14:08:53 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/5.5.36
Link: <http://kefo.no/wp-json/>; rel="https://api.w.org/", <https://wp.me/92qpF>; rel=shortlink

[k] [Connected] >
```

Figur 3: Klient get request

Dersom det er feil FQDN ,så vil serveren sende en RST istede for ACK for datane og en kan spørre på nytt i fra klienten.

## 2.5 Skisse av kommunikasjon



Figur 4: Klient WPS Kommunikasjon