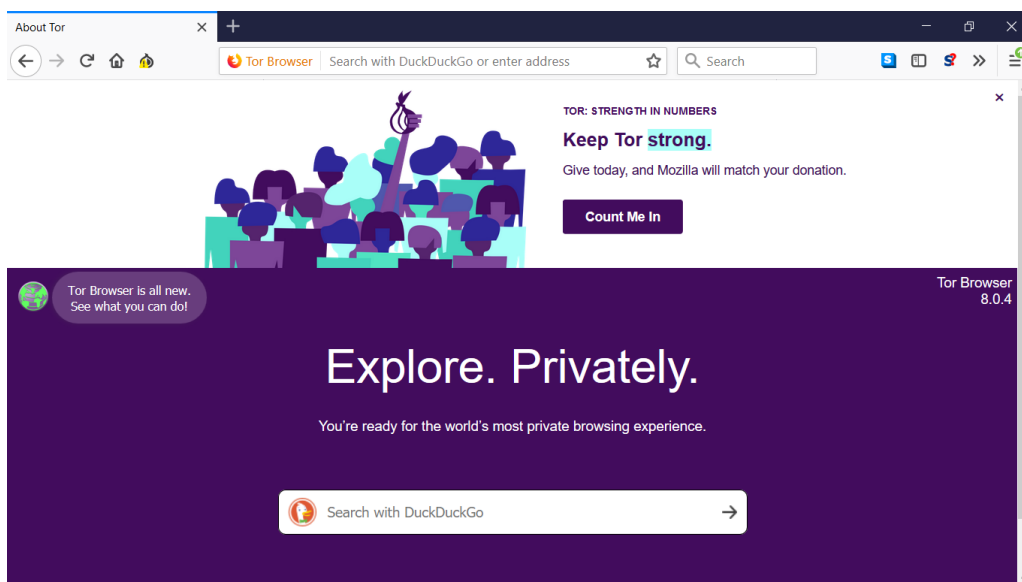


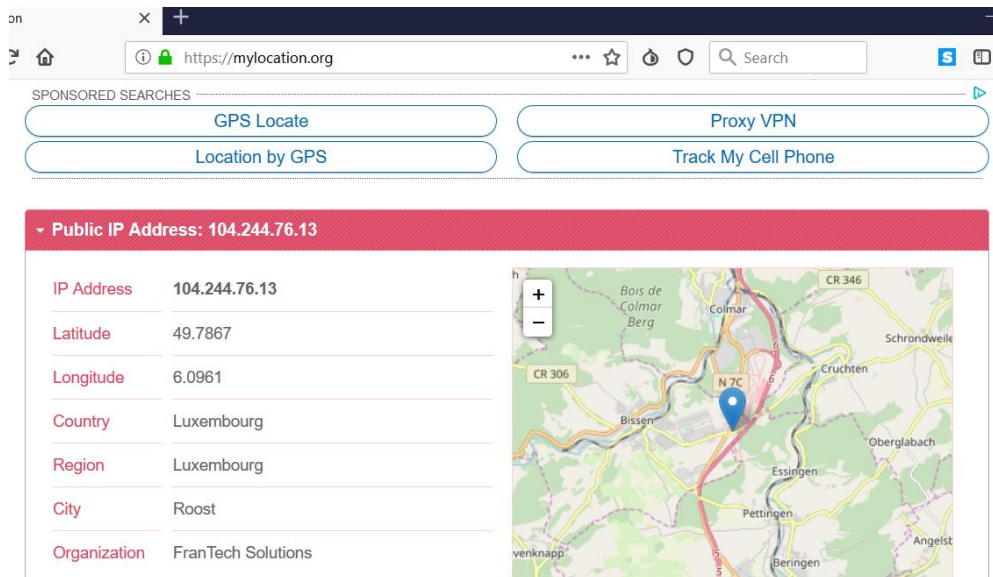
5 Implementacija skrivenog servisa

U ovom delu će biti opisan način na koji je moguće konfigurisati sopsveni onion servis¹.

Najpre, potrebno je instalirati Tor pretraživač². Na slici 5.1 je prikazana početna strana Tor pretraživača a na slici 5.2 dodeljena lokacija i IP adresa, odnosno lokacija i IP adresa izlaznog čvora.



Slika 5.1 Početna strana Tor pretraživača



Slika 5.2 Lokacija i IP adresa

¹ Servis je pokrenut na operativnom sistemu Windows 10

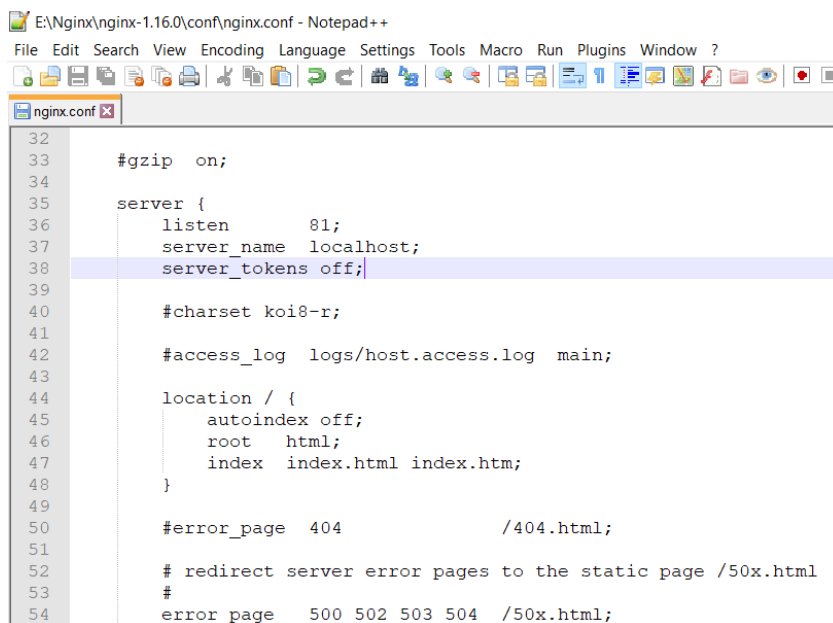
² <https://2019.www.torproject.org/projects/torbrowser.html.en>

Sledeći korak je instaliranje lokalnog web servera. Preporučuje se nginx³ ili lighttpd⁴. Apache server nije najbolja opcija za anonimnost koja se zahteva. U ovom radu je korišćen nginx server. Prilikom instaliranja i konfiguirisanja nginx servera (ili servera koji korisnici odaberu da koriste), korisnici bi trebalo da vode računa da ne odaju svoju IP adresu (ili druge informacije koje mogu odati identitet korisnika) zbog loše konfigurisanog servera. Trebalo bi da se pridržavaju nekih osnovnih pravila [40]:

- Web server treba slušati samo localhost, odnosno 127.0.0.1
- Treba se onemogućiti nabranje direktorijuma
- Treba se onemogućiti potpis servera (ime i verzija)
- Treba se onemogućiti izveštavanje grešaka

Javna IP adresa nikad ne bi trebalo da bude otkrivena. Nabranje direktorijuma može potencijalno otkriti osetljive informacije napadaču. Izveštavanje grešaka zavisi od backend jezika koji se koristi (PHP, NodeJS, Python, itd.), i može da otkrije poverljive informacije o korisniku, stoga se treba onemogućiti. Na kraju, treba se voditi računa o samom kodu u kom je pisan servis, odnosno da li se u nekom delu otkrivaju informacije koje bi trebalo ostati sakrivene.

Na slici 5.3 prikazana je konfiguracija nginx servera, gde se vidi da server sluša localhost, da je potpis servera onemogućen (`server_tokens off`) [39] i da je listanje direktorijuma onemogućeno (`autoindex off`).



```
32 #gzip on;
33
34
35 server {
36     listen      81;
37     server_name localhost;
38     server_tokens off;
39
40     #charset koi8-r;
41
42     #access_log logs/host.access.log main;
43
44     location / {
45         autoindex off;
46         root    html;
47         index  index.html index.htm;
48     }
49
50     #error_page 404              /404.html;
51
52     # redirect server error pages to the static page /50x.html
53     #
54     error_page 500 502 503 504 /50x.html;
```

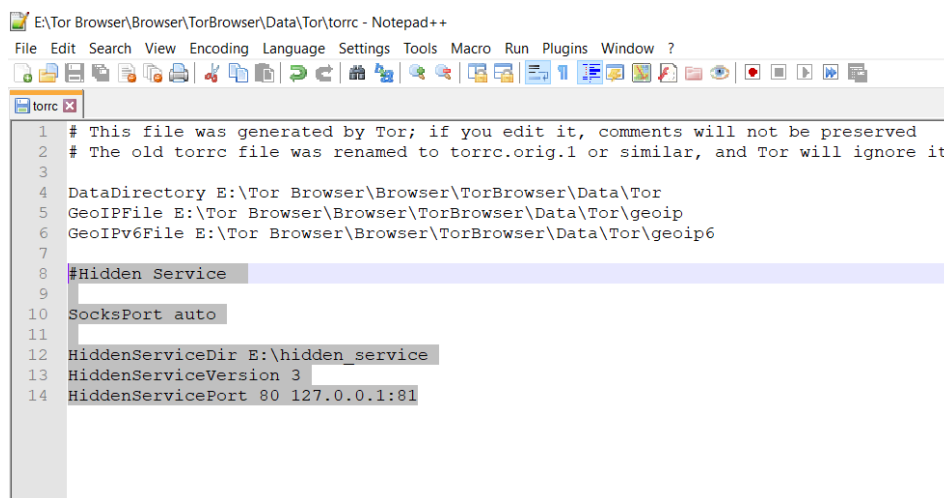
Slika 5.3 Konfiguracija nginx servera

³ <http://nginx.org/en/download.html>

⁴ <https://www.lighttpd.net/download/>

Sledeći korak je konfigurisanje onion servisa. Za to treba otvoriti torrc fajl koji se nalazi u na lokaciji "...\\Browser\\TorBrowser\\Data\\Tor", gde je "..." folder gde je instalran Tor pretraživač. Za servis je potrebno uneti polja *HiddenServiceVersion*, *HiddenServiceDir* i *HiddenServicePort*. *HiddenServiceDir* predstavlja direktorijum u kom će Tor generisati potrebne informacije o onion servisu. *HiddenServicePort* predstavlja virtualni port (za koji će korisnici koji koriste servis misliti da koriste), i IP adresu i port koji će preusmeriti vezu na ovaj virtualni port. *HiddenServiceVersion* će imati vrednost 3 (u ovom radu se implementiraju onion servisi verzije 3 umesto verzije 2) [38].

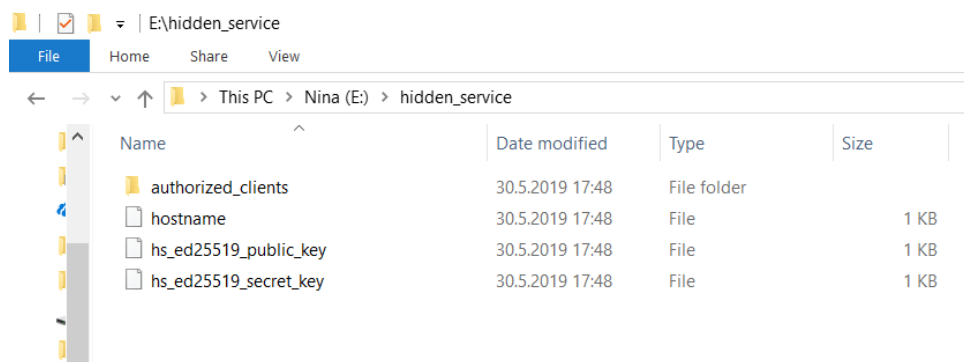
Na slici 5.4 prikazana je konfiguracija torrc fajla.



```
1 # This file was generated by Tor; if you edit it, comments will not be preserved
2 # The old torrc file was renamed to torrc.orig.1 or similar, and Tor will ignore it
3
4 DataDirectory E:\\Tor Browser\\Browser\\TorBrowser\\Data\\Tor
5 GeoIPFile E:\\Tor Browser\\Browser\\TorBrowser\\Data\\Tor\\geoip
6 GeoIPv6File E:\\Tor Browser\\Browser\\TorBrowser\\Data\\Tor\\geoip6
7
8 #Hidden Service
9
10 SocksPort auto
11
12 HiddenServiceDir E:\\hidden_service
13 HiddenServiceVersion 3
14 HiddenServicePort 80 127.0.0.1:81
```

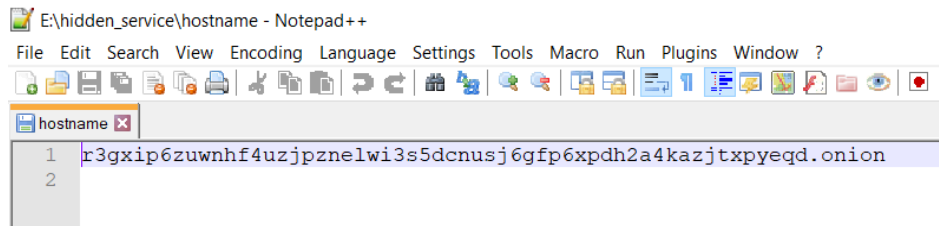
Slika 5.4 Konfiguracija torrc fajla

Nakon menjanja konfiguracije torrc fajla, potrebno je pokrenuti Tor pretraživač (ako je bio pokrenut ranije treba ga restartovati). Tor pretraživač će tada generisati adresu, odnosno ime servisa (master identifikacioni ključ) i potrebne ključeve i smestiti te informacije u direktorijum koji je naveden u polju *HiddenServiceDir*. Sadržaj ovog direktorijuma se može videti na slici 5.5.



Slika 5.5 Direktorijum gde su generisani ključevi i ime servisa

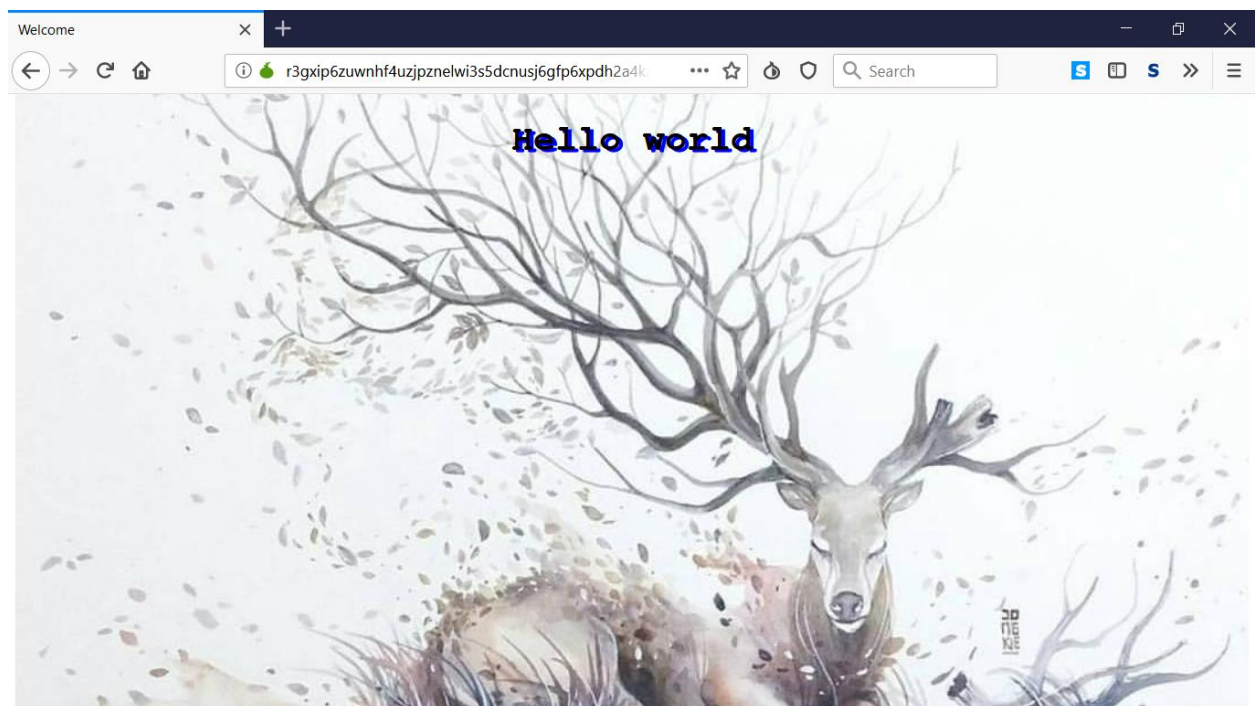
Ime servisa se može naći u fajlu *hostname*. Ovaj fajl je prikazan na slici 5.6.



Slika 5.6 Generisano ime servisa

Da bi potencijalni korisnici saznali za ovaj skriveni servis trebaju da saznaju njegovu adresu. Korisnicima se direktno može preneti informacija o adresi, može se reklamirati na popularnim sajtovima koji sadrže linkove ka skrivenim servisima kao što je Hidden Wiki, ili se može reklamirati na društvenim mrežama i forumima.

Ime iz fajla *hostname* treba prekopirati u Tor pretraživaču, i, ako je sve u redu, servis će biti dostupan. Na slici 5.7 je prikazan skriveni servis.

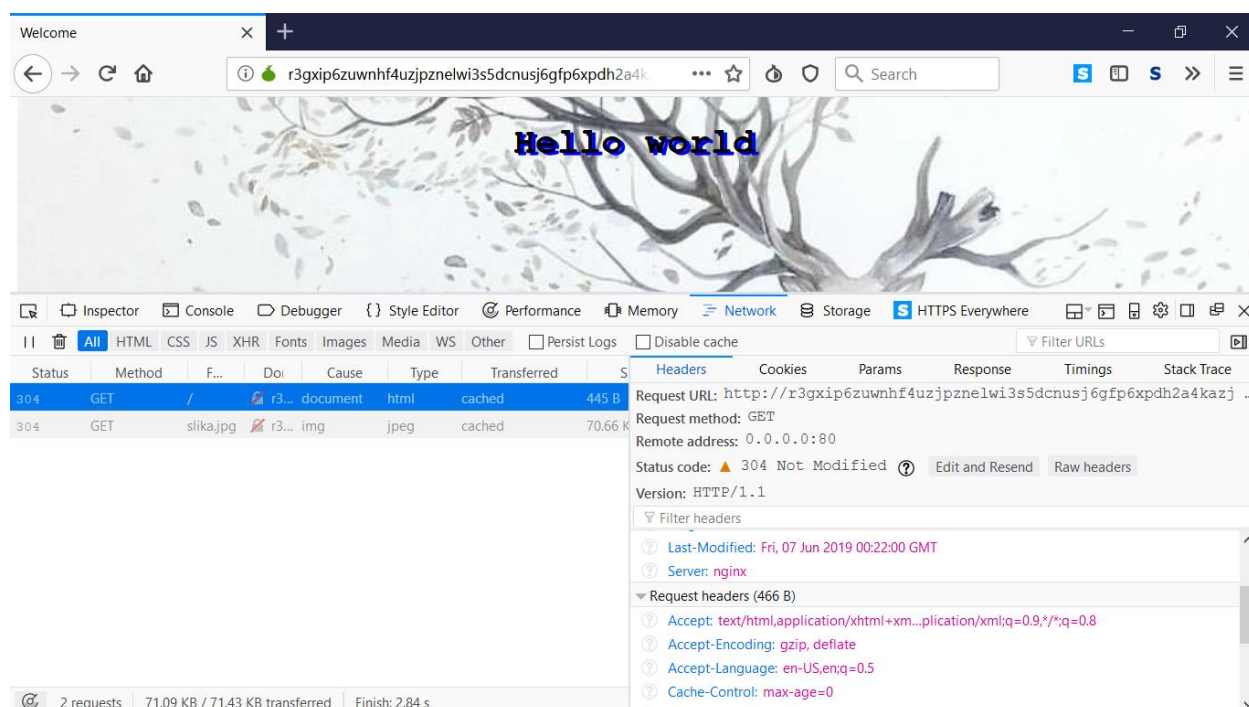


Slika 5.7 Skrivena web strana

U ovom radu skriveni servis je jednostavna web strana. Naravno, korisnik može postaviti svoju online prodavnicu, magazin, web sajt, itd.

Servis će biti dotupan onoliko vremena koliko server i Tor pretraživač rade.

Kada se provere HTTP zaglavlja, vidi se da IP adresa nije otkrivena. Slika 5.8 to ilustruje. Za udaljenu adresu prikazuje se 0.0.0.0:80, gde je port 80 virtualan port koji je naveden u torrc fajlu.



Slika 5.8 HTTP zaglavlje

Dalje analiziranje paketa izvršili smo pomoću analizatora mrežnih protokola - Wireshark⁵ koji koristi biblioteku za praćenje paketa - Npcap⁶. Pomoću Wireshark-a i Npcap-a moguće je pratiti loopback pakete (pakete na adresi 127.0.0.1).

Ovu analizu paketa smo započeli tako što smo pristupili servisu preko Tor pretraživača instaliranog na mobilnom telefonu, gde je operativni sistem Android. Za vreme pristupanja servisu Wireshark je sačuvao i prikazao sve loopback pakete koji se mogu videti na slikama 5.9, 5.10, 5.11 i 5.12. Zelenom bojom zabeleženi su HTTP zahtevi i odgovori, a sivom TCP paketi. Na slici 5.9 je prikazan HTTP GET zahtev koji dolazi sa telefona. Server sluša na portu 81. Klijent sluša na portu 51554. Server šalje HTTP OK odgovor i šalje html stranu, za šta mu trebaju dva TCP paketa (jedan za potvrdu primanja zahteva - ACK, i drugi za sadržaj html strane), što je prikazano na slici 5.10. Klijent onda zahteva pozadinsku sliku, što se vidi na slici 5.11. Nakon što je poslao sliku klijentu, za šta mu je potrebno 137 TCP paketa, server onda šalje HTTP OK odgovor koji možemo videti na slici 5.12.

⁵ <https://www.wireshark.org/download.html>

⁶ <https://nmap.org/npcap/>

*Npcap Loopback Adapter

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 0

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	66	51554 → 81 [SYN] Seq=0 Win=64240 Len=0 MSS=65495 WS=256 SACK_PER...
2	0.000311	127.0.0.1	127.0.0.1	TCP	66	81 → 51554 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=65495 WS=25...
3	0.000585	127.0.0.1	127.0.0.1	TCP	54	51554 → 81 [ACK] Seq=1 Ack=1 Win=525568 Len=0
16	0.480455	127.0.0.1	127.0.0.1	HTTP	430	GET / HTTP/1.1
17	0.480563	127.0.0.1	127.0.0.1	TCP	54	81 → 51554 [ACK] Seq=1 Ack=377 Win=65536 Len=0
18	0.482364	127.0.0.1	127.0.0.1	TCP	590	81 → 51554 [ACK] Seq=1 Ack=377 Win=65536 Len=536 [TCP segment of...
19	0.482382	127.0.0.1	127.0.0.1	HTTP	194	HTTP/1.1 200 OK (text/html)
20	0.482498	127.0.0.1	127.0.0.1	TCP	54	51554 → 81 [ACK] Seq=377 Ack=677 Win=524800 Len=0
67	1.326333	127.0.0.1	127.0.0.1	HTTP	430	GET /slika.jpg HTTP/1.1
68	1.326447	127.0.0.1	127.0.0.1	TCP	54	81 → 51554 [ACK] Seq=677 Ack=753 Win=65792 Len=0

> Frame 16: 430 bytes on wire (3440 bits), 430 bytes captured (3440 bits) on interface 0

> Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 51554, Dst Port: 81, Seq: 1, Ack: 1, Len: 376

> Hypertext Transfer Protocol

> GET / HTTP/1.1\r\n

Host: r3gxp6zuwnhf4uzjpnzlw3s5dcnusj6gfp6xpdh2a4kazjtxpyeqd.onion\r\n

User-Agent: Mozilla/5.0 (Android 6.0; Mobile; rv:60.0) Gecko/20100101 Firefox/60.0\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n

Accept-Language: en-GB,en-US;q=0.7,en;q=0.3\r\n

Accept-Encoding: gzip, deflate\r\n

Connection: keep-alive\r\n

Upgrade-Insecure-Requests: 1\r\n

\r\n

[Full request URI: <http://r3gxp6zuwnhf4uzjpnzlw3s5dcnusj6gfp6xpdh2a4kazjtxpyeqd.onion/>]

[HTTP request 1/2]

Slika 5.9 HTTP GET zahtev

*Npcap Loopback Adapter

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 0

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	66	51554 → 81 [SYN] Seq=0 Win=64240 Len=0 MSS=65495 WS=256 SACK_PER...
2	0.000311	127.0.0.1	127.0.0.1	TCP	66	81 → 51554 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=65495 WS=25...
3	0.000585	127.0.0.1	127.0.0.1	TCP	54	51554 → 81 [ACK] Seq=1 Ack=1 Win=525568 Len=0
16	0.480455	127.0.0.1	127.0.0.1	HTTP	430	GET / HTTP/1.1
17	0.480563	127.0.0.1	127.0.0.1	TCP	54	81 → 51554 [ACK] Seq=1 Ack=377 Win=65536 Len=0
18	0.482364	127.0.0.1	127.0.0.1	TCP	590	81 → 51554 [ACK] Seq=1 Ack=377 Win=65536 Len=536 [TCP segment of...
19	0.482382	127.0.0.1	127.0.0.1	HTTP	194	HTTP/1.1 200 OK (text/html)
20	0.482498	127.0.0.1	127.0.0.1	TCP	54	51554 → 81 [ACK] Seq=377 Ack=677 Win=524800 Len=0
67	1.326333	127.0.0.1	127.0.0.1	HTTP	430	GET /slika.jpg HTTP/1.1
68	1.326447	127.0.0.1	127.0.0.1	TCP	54	81 → 51554 [ACK] Seq=677 Ack=753 Win=65792 Len=0

> Frame 19: 194 bytes on wire (1552 bits), 194 bytes captured (1552 bits) on interface 0

> Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 81, Dst Port: 51554, Seq: 537, Ack: 377, Len: 140

> [2 Reassembled TCP Segments (676 bytes): #18(536), #19(140)]

> Hypertext Transfer Protocol

> HTTP/1.1 200 OK\r\n

Server: nginx\r\n

Date: Fri, 07 Jun 2019 00:38:48 GMT\r\n

Content-Type: text/html\r\n

> Content-Length: 445\r\n

Last-Modified: Fri, 07 Jun 2019 00:22:00 GMT\r\n

Connection: keep-alive\r\n

ETag: "5cf9ae28-1bd"\r\n

Accept-Ranges: bytes\r\n

\r\n

Slika 5.10 HTTP OK odgovor

*Npcap Loopback Adapter

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 0

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	66	51554 → 81 [SYN] Seq=0 Win=64240 Len=0 MSS=65495 WS=256 SACK_PER...
2	0.000311	127.0.0.1	127.0.0.1	TCP	66	81 → 51554 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=65495 WS=25...
3	0.000585	127.0.0.1	127.0.0.1	TCP	54	51554 → 81 [ACK] Seq=1 Ack=1 Win=525568 Len=0
16	0.480455	127.0.0.1	127.0.0.1	HTTP	430	GET / HTTP/1.1
17	0.480563	127.0.0.1	127.0.0.1	TCP	54	81 → 51554 [ACK] Seq=1 Ack=377 Win=65536 Len=0
18	0.482364	127.0.0.1	127.0.0.1	TCP	590	81 → 51554 [ACK] Seq=1 Ack=377 Win=65536 Len=536 [TCP segment of...
19	0.482382	127.0.0.1	127.0.0.1	HTTP	194	HTTP/1.1 200 OK (text/html)
20	0.482498	127.0.0.1	127.0.0.1	TCP	54	51554 → 81 [ACK] Seq=377 Ack=677 Win=524800 Len=0
67	1.326333	127.0.0.1	127.0.0.1	HTTP	430	GET /slika.jpg HTTP/1.1
68	1.326447	127.0.0.1	127.0.0.1	TCP	54	81 → 51554 [ACK] Seq=677 Ack=753 Win=65792 Len=0
69	1.405958	127.0.0.1	127.0.0.1	TCP	590	81 → 51554 [ACK] Seq=677 Ack=753 Win=65792 Len=536 [TCP segment ...

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 51554, Dst Port: 81, Seq: 377, Ack: 677, Len: 376

Hypertext Transfer Protocol

GET /slika.jpg HTTP/1.1\r\n

Host: r3gxp6zuwnhf4uzjpnznelwi3s5dcnusj6gfp6xpdh2a4kazjtxpyeqd.onion\r\n

User-Agent: Mozilla/5.0 (Android 6.0; Mobile; rv:60.0) Gecko/20100101 Firefox/60.0\r\n

Accept: */*\r\n

Accept-Language: en-GB,en-US;q=0.7,en;q=0.3\r\n

Accept-Encoding: gzip, deflate\r\n

Referer: http://r3gxp6zuwnhf4uzjpnznelwi3s5dcnusj6gfp6xpdh2a4kazjtxpyeqd.onion/\r\n

Connection: keep-alive\r\n

\r\n

[Full request URI: http://r3gxp6zuwnhf4uzjpnznelwi3s5dcnusj6gfp6xpdh2a4kazjtxpyeqd.onion/slika.jpg]

[HTTP request 2/2]

[Prev request in frame: 16]

Slika 5.11 HTTP GET zahtev za sliku

*Npcap Loopback Adapter

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 0

No.	Time	Source	Destination	Protocol	Length	Info
219	1.448135	127.0.0.1	127.0.0.1	TCP	590	81 → 51554 [ACK] Seq=68593 Ack=753 Win=65792 Len=536 [TCP segmen...
220	1.448148	127.0.0.1	127.0.0.1	TCP	590	81 → 51554 [ACK] Seq=69129 Ack=753 Win=65792 Len=536 [TCP segmen...
221	1.448160	127.0.0.1	127.0.0.1	TCP	590	81 → 51554 [ACK] Seq=69665 Ack=753 Win=65792 Len=536 [TCP segmen...
222	1.448172	127.0.0.1	127.0.0.1	TCP	590	81 → 51554 [ACK] Seq=70201 Ack=753 Win=65792 Len=536 [TCP segmen...
223	1.448183	127.0.0.1	127.0.0.1	TCP	590	81 → 51554 [ACK] Seq=70737 Ack=753 Win=65792 Len=536 [TCP segmen...
224	1.448194	127.0.0.1	127.0.0.1	TCP	590	81 → 51554 [ACK] Seq=71273 Ack=753 Win=65792 Len=536 [TCP segmen...
225	1.448206	127.0.0.1	127.0.0.1	TCP	590	81 → 51554 [ACK] Seq=71809 Ack=753 Win=65792 Len=536 [TCP segmen...
226	1.448217	127.0.0.1	127.0.0.1	TCP	590	81 → 51554 [ACK] Seq=72345 Ack=753 Win=65792 Len=536 [TCP segmen...
227	1.448229	127.0.0.1	127.0.0.1	HTTP	440	HTTP/1.1 200 OK (JPEG JFIF image)
228	1.448430	127.0.0.1	127.0.0.1	TCP	54	51554 → 81 [ACK] Seq=753 Ack=69921 Win=525568 Len=0
229	1.448443	127.0.0.1	127.0.0.1	TCP	54	51554 → 81 [ACK] Seq=753 Ack=73267 Win=525568 Len=0

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 81, Dst Port: 51554, Seq: 72881, Ack: 753, Len: 386

[137 Reassembled TCP Segments (72590 bytes): #69(536), #70(536), #71(536), #72(536), #73(536), #74(536), #75(536), #76(536), #77(536), #78(536),

Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\n

Server: nginx\r\n

Date: Fri, 07 Jun 2019 00:38:49 GMT\r\n

Content-Type: image/jpeg\r\n

Content-Length: 72354\r\n

Last-Modified: Fri, 06 May 2016 10:57:04 GMT\r\n

Connection: keep-alive\r\n

ETag: "572c7880-11aa2"\r\n

Accept-Ranges: bytes\r\n

\r\n

Slika 5.12 HTTP OK odgovor za sliku

Dodatno, može se omogućiti autorizacija klijenata [38], [41]. Za to će se za svakog autorizovanog klijenta, u folderu `authorized_clients` koji se nalazi u direktorijumu koji je naveden u polju `HiddenServiceDir`, navesti ključ, u fajlu sa sufiksom `".auth"`, na primer *ime_fajla.auth*, gde je ime_fajla bilo koje ime.

Format mora da bude:

`<auth-type>:<key-type>:<base32-encoded-public-key>`,

gde je `auth-type` - `"descriptor"`, `key-type` - `"x25519"`, i `base32-encoded-public-key` - base32 reprezentacija autorizacijskog ključa klijenta.

Svaki fajl mora sadržati samo jednu liniju. Klijentska autorizacija će biti omogućena ako se Tor pretraživač pokrene bar sa jednim ovakvim fajlom. Sada, niko osim odabranih klijenta neće imati pristup servisu. Ako se nekom klijentu treba ukinuti dozvola pristupa, jednostavno se briše njegov `.auth` fajl i restartuje se Tor pretraživač.

Za sada ove autorizacijske ključeve klijenta je potrebno generisati samostalno. To se može uraditi pomoću postojećih skripti⁷.

Da bi ovlašćeni klijent mogao pristupiti skrivenom servisu mora imati *ClientOnionAuthDir* polje u `torrc` fajlu, u kome će navesti put do direktorijuma gde čuva svoj ključ. Ključ mora biti fajl sa sufiksom `".auth_private"`, na primer *ime_fajla.auth_private*, gde je ime_fajla bilo koje ime.

Format mora da bude:

`<onion-address>:descriptor:x25519:<base32-encoded-privkey>`

gde je `onion-address` - onion adresa bez `".onion"` dela i `base32-encoded-privkey` - base32 reprezentacija autorizacijskog ključa klijenta.

⁷ <https://gist.github.com/mtigas/9c2386adf65345be34045dace134140b> i <https://github.com/haxxpop/torkeygen>