Incentivize decentralized WiFi roaming through VPN on home routers

RP2 rp-id 25, Security and Network Engineering, UvA

Sander.Lentink@os3.nl Peter.Boers@surfnet.nl

2019-11-13

Introduction

We desire Wi-Fi

- Wi-Fi being "the best technology for Mobile Data Offloading (MDO)" (Gupta and Rohil 2012)

Enabling Wi-Fi problematic

- concerns around security, violating terms / illegal content (Schneier 2008)
- laws prevent municipality provided "free WiFi" (Chamberlain 2019)
- telecommunications lobby against new projects (Gurley and O'Shaughnessy 2019)

When we access Wi-Fi

- users unaware of privacy risks (Consolvo et al. 2010)
- Free WiFi: captive portal

Intro: Overcome mutual trust issue

Client tunnels via home router (Sastry, Crowcroft, and Sollins 2007)

```
 --------                           -----------
|Client  |  |foreign|  |internet|  |Client's   |
|  ______|  |  AP   |  | (WAN)  |  |home AP    |
| |VPN   |  |_______|  |________|  |__________ |
| |Client|------------------------>|VPN server||
|_|______|  |-------|  |--------|  |__________||
```

- Client has no privacy leaks
- Wi-Fi AP[1] provider has no liability worries

[1] Access Point

Intro: example setup

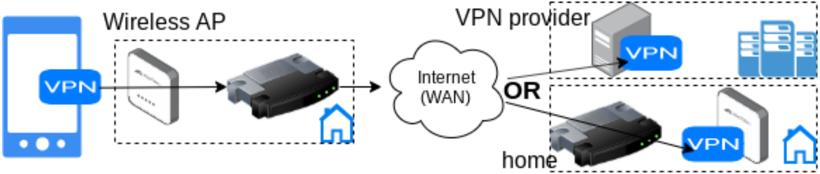


Figure 1: “Client connects to VPN endpoint via foreign AP”

Intro: Research Question

Can we design a protocol — using existing protocols available on COTS (commercially off the shelf) clients — that eliminates the need for trust between client and Wi-Fi provider, using a VPN tunnel?

Intro: Sub Questions

- Enforce network policies?
- Validate if VPN server listens on endpoint?
- Client communicate VPN endpoint to AP?
- Modify authentication (802.1x) server to enable this protocol?
- Verify protocol: Proof of Concept (PoC)?

Intro: Questions TL;DR

- Design Protocol
- Test with PoC

Intro: Related solutions

Closed options

- Ad based: World Wi-Fi
- Education Roaming: Eduroam
- Government Roaming: Govroam
- Share WiFi, earn points/data/credits: Karma
- Home router managed by provider: KPN's Fon
- Paid / broker based: Tmobile/Vodafone hotspots

Open solutions

- Open Wireless Movement, backed by Electronic Frontier Foundation

Methodology

- Example flow: overview of concept
- 802.1x EAP identity
- Protocol in authentication server

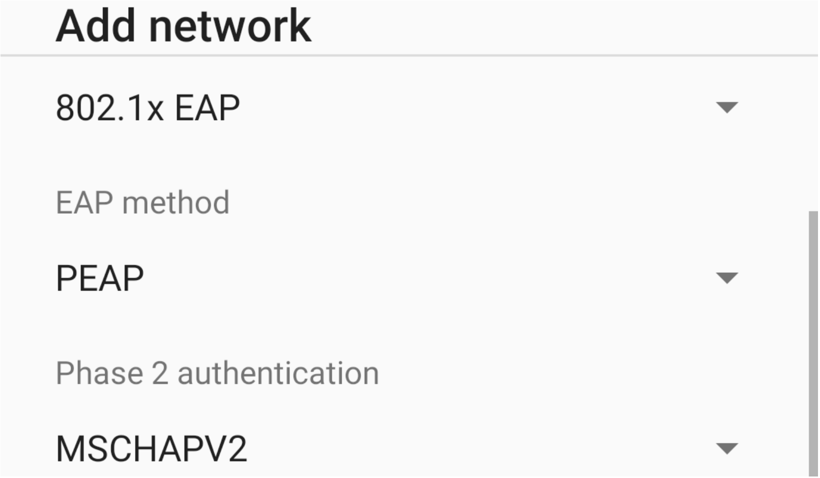


Figure 2: Extensible Authentication Protocol

Method: example flow 1/3

```
                   AP (SoC) <-----L2----> router
         _____________|_____________   _____|____
client  |hostapd  auth   pre   filter| |DHCP   WAN|   VPN
   |        |      |      |      |      |      |      |
   a        |      |      |      |      |      |      |
   #---b--->#      |      |      |      |      |      |
   #        #--c-->#      |      |      |      |      |
   #        #      #--d-->#      |      |      |      |
   #        #      #      #--e----------------------->#
   #        #      #      f      |      |      |      |
   #        #      #      #--g-->#      |      |      |
   #        #      #<--h--#      |      |      |      |
   #        #<--i--#      |      |      |      |      |
   #<---j---#      |      |      |      |      |      |
   #---k-------------------------=----->#      |      |
   #<--------------------------------l--#      |      |
   #---m-------------------------=------------------->#
```

Method: example flow 2/3

```
                        AP (SoC) <-----L2----> router
              _____________|_____________   _____|____
     client  |hostapd  auth   pre  filter| |DHCP   WAN|   VPN
        |        |      |      |      |      |      |      |
        a        |      |      |      |      |      |      |
        #---b--->#      |      |      |      |      |      |
        #        #--c-->#      |      |      |      |      |
        #        #      #--d-->#      |      |      |      |
        #        #      #      #--e----------------------->#
```

a. client (supplicant) scans for AP, finds foreign AP with SSID of protocol
b. supplicant => authenticator (`hostapd`), VPN endpoint location in 802.1x identity
c. authenticator => authentication server
d. authentication server => custom `pre`-authorize script
e. provided info points to a VPN server?

Method: example flow 3/3

```
client  |hostapd  auth   pre  filter| |DHCP   WAN|   VPN
   #        #      #      f--g-->#      |      |      |
   #        #      #<--h--#      |      |      |      |
   #        #<--i--#      |      |      |      |      |
   #<---j---#      |      |      |      |      |      |
   #---k-------------------------=----->#      |      |
   #<--------------------------------l--#      |      |
   #---m-------------------------=------------------->#
```

f. `if VPN: continue else return` 802.1x rejected
g. allow (whitelist) egress for provided VPN details
h. OK
i. OK
j. 802.1x client accepted (`wlan` bridged (L2) with `eth0`)
k. client requests DHCP lease (IP address)
l. router provides IP to client (thus NAT* in router)
m. client => VPN server

* Network Address Translation

Method: example flow TL;DR

- SoC connected to router =
  - VPN server
  - Wi-Fi AP
  - Authentication server
- When your phone finds foreign AP
  - AP whitelists VPN server
  - phone uses VPN

Method: Client; VPN server

- Out of scope

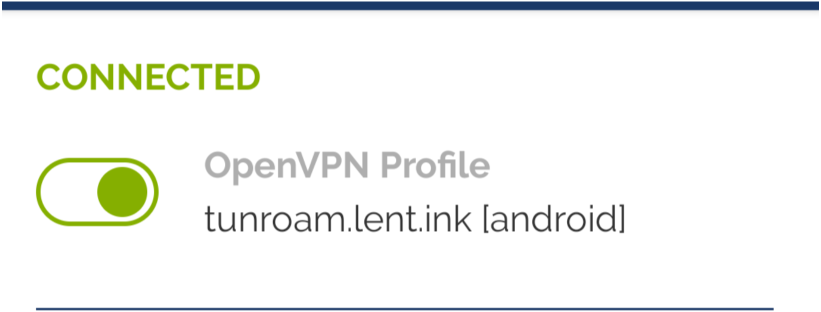
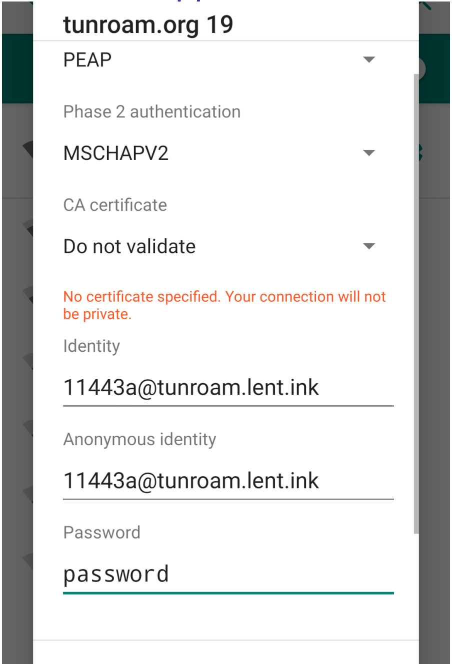


Figure 3: VPN client on Android

Method: Client; 802.1x supplicant



Method: 802.1x identities

VPN ports + flags + delimiter (`@`) + realm (hostname or IP)

`32_33_2f_06443_11443  a  @  10.10.10.10`

Anonymous id (`anonid`)

Proxying server

Regular id (`innerid`)

Inside TLS tunnel (**Protected**-EAP)

Method: IP Protocols

IP protocol + additional value (port)

`32_33_2f_06443_11443a@10.10.10.10`

IP protocol	ID
TCP (Transmission Control)	0x06
UDP (User Datagram)	0x11
GRE (Generic Routing Encapsulation)	0x2F
ESP (Encap Security Payload)	0x32
AH (Authentication Header)	0x33

Method: pre-authorize

```
$ validate_anonid.py 11443_06443_00testA@tunroam.lent.ink
WARNING the additional value is not a port number
INFO suggesting whitelist rules
{ 'iptables-nft -A OUTPUT -j ACCEPT -d tunroam.lent.ink \
     --protocol 17 --dport 443',
  'iptables-nft -A OUTPUT -j ACCEPT -d tunroam.lent.ink \
     --protocol 6 --dport 443' }
INFO Welcome aboard 11443_06443_00testA@localhost
```

- VPN endpoint validation
- Network policies

Method: Network requirements

> *TUN works with IP frames. TAP works with Ethernet frames.*[2]

Shared SSID

Like Eduroam / Govroam: TUNroam;

`tunroam.org 19`

- Version number indicates client requirements (20*19*)

[2] https://www.kernel.org/doc/Documentation/networking/tuntap.txt

Method: Additional network traffic?

Local scope

- Network management (e.g. ARP[3])

Leaking to Internet Service Provider (ISP)

- DNS

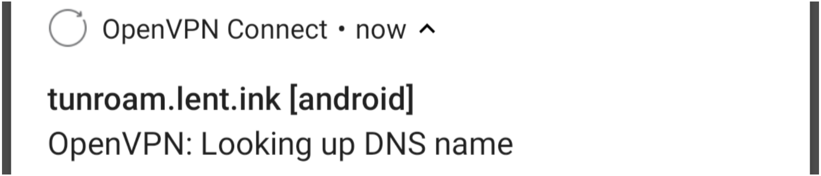


Figure 5: VPN endpoint discovery by client

[3] Address Resolution Protocol

Method: DNS

AP provider doesn't want DNS logged by ISP

Required: specific subdomain

```
iptables-nft -I OUTPUT -j ALLOW --algo bm \
  -p udp --dport 53 \
  --match string --hex-string "|07|tunroam|"
```

Method: System on Chip SoC

Test setup

RPi

- Raspbian

```
cat /proc/cpuinfo|grep Model
Model           : Raspberry Pi 3 Model B Rev 1.2
```

Entry level setup

- Armbian
- Orange Pi Zero Plus (1000M Ethernet, 512MB RAM, onboard WiFi)
- OPi + MicroSD + USB cable & power = 20EU[4]

[4] excl. shipping

Results

- Protocol defined
- Protocol (partially) implemented
  - PoC doing NAT
  - Identity validation
  - VPN endpoint validation

Discussion

TUNroam

Pro

- client:
  - privacy through VPN on any network
  - More free Wi-Fi locations
  - No captive portal
- AP:
  - Open source
  - Liability
  - Decentralized: nobody controls it

Con

- Decentralized: no financial incentive to join/promote
- Provider routers != Open(Wrt)
- VPN
  - Latency
  - Bandwidth

Discuss: Potential APs:

- shared office space/housing
- home router
- current open Wi-Fi

Discuss: Future work

Missing in PoC

- Proxying RADIUS request

Suggestions

- Bandwidth management
- Enforce network policies
- IPv6
- Home != fixed IP: Dynamic DNS

Demo

```
PEAP, MS-CHAPv2, "password"
```

Please connect to SSID “tunroam.org 19”

```
# OpenVPN, TCP/UDP 443
06443_11443_00testA@tunroam.lent.ink
```

Questions?

- Get involved at github.com/tunroam
- Reach me at linkedin.com/in/svlentink

Appendix: bonus slides

Slides to help answer possible questions.

And things that didn't fit due to time constraints.

Appendix: tests using fast.com

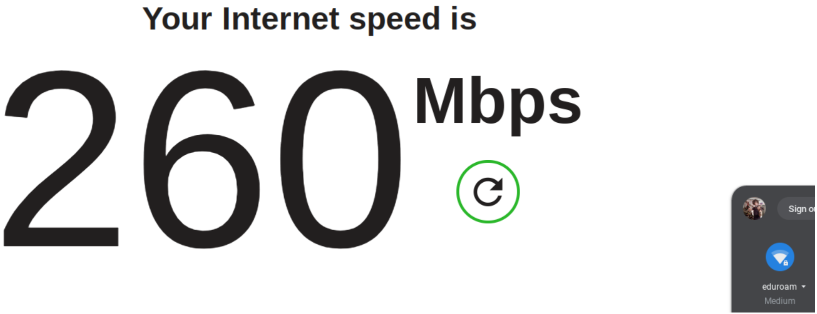


Figure 6: Eduroam network Surfnet office

Appendix: tests using fast.com

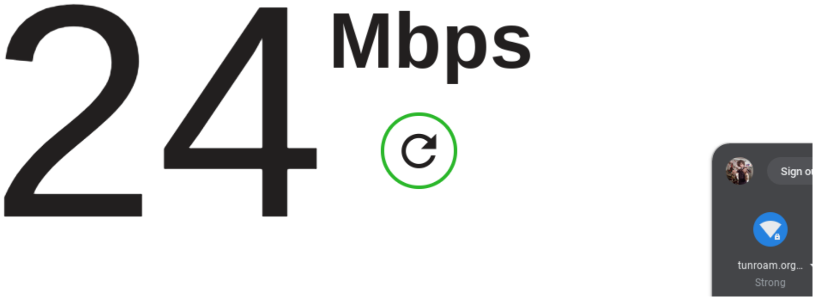


Figure 7: OrangePi doing NAT

Appendix: Covert channel? Abuse?

Using VPN is easier due to:

- Limited DNS requests
- Only one IP address
- Limited ports

Appendix: Bridge vs. NAT

Bridge

- Sequence diagram = bridged (home setup)
- Avoid double NAT
- Avoid NAT in software

Network Address Translation

- NAT works everywhere
- PoC/Demo = NAT

Multiple APs (Campus / Airport)

- Authentication server separate
- Network policies

Appendix: RADIUS proxying

```
$ ls /etc/freeradius/*/sites-enabled
default  inner-tunnel
```

```
$ ss -4lpun|grep -E "(1812|Port)"
State  Recv-Q Send-Q Local Address:Port   Peer Address:Port
UNCONN 0      0            0.0.0.0:1812        0.0.0.0:*
UNCONN 0      0          127.0.0.1:18120       0.0.0.0:*
```

- Inner does CHAP

Appendix: Challenge-Handshake Authentication Protocol

Microsoft CHAP v2

Authentication server: proxy-server

```
if valid_vpn_endpoint and valid_anonid: # anonymous identi
```

Authentication server: inner-tunnel

```
  return RLM_MODULE_OK, (), \
    ( ('Cleartext-Password', 'password'), )
```

Appendix: VPN protocols

Initial

- Which VPN protocol(s) fit in the protocol?
- What attributes do we need to validate to determine if a VPN server is listening on an endpoint?

Different approach

- Stealth VPN servers
- IP protocols
- Check socket
- Allow evolution

Appendix: Flag character

bit	name	description
0000?	`validate_certificate`	Validate 802.1x certificate?
000?0	RESERVED	
00?00	RESERVED	
0?000	RESERVED	
?0000	RESERVED	

- base32 character: RFC4648

Well-known ports

AP MAY filter well-known ports (below 1024)

- except:
  - 22 (socks tunnel),
  - 443 (HTTPS tunnel)
  - 500 (IKE[5] for IPsec)

[5] Internet Key Exchange

References

Chamberlain, Kendra. 2019. "Municipal Broadband Is Roadblocked or Outlawed in 26 States." https://broadbandnow.com/report/municipal-broadband-roadblocks/.

Consolvo, Sunny, Jaeyeon Jung, Ben Greenstein, Pauline Powledge, Gabriel Maganis, and Daniel Avrahami. 2010. "The Wi-Fi Privacy Ticker: Improving Awareness & Control of Personal Information Exposure on Wi-Fi." In *Proceedings of the 12th Acm International Conference on Ubiquitous Computing*, 321–30. UbiComp '10. New York, NY, USA: ACM. https://doi.org/10.1145/1864349.1864398.

Gupta, Vishal, and Mukesh Kumar Rohil. 2012. "Enhancing Wi-Fi with Ieee 802.11 U for Mobile Data Offloading." *International Journal of Mobile Network Communications & Telematics (IJMNCT)* 2 (4): 19–29. https://www.researchgate.net/profile/Mukesh_Rohil/publication/268258653_Enhancing_Wi-Fi_with_IEEE_80211u_for_Mobile_Data_Offloading/links/54d1b4a80cf28370d0e0ff56/Enhancing-Wi-Fi-with-IEEE-80211u-for-Mobile-Data-Offloading.pdf.