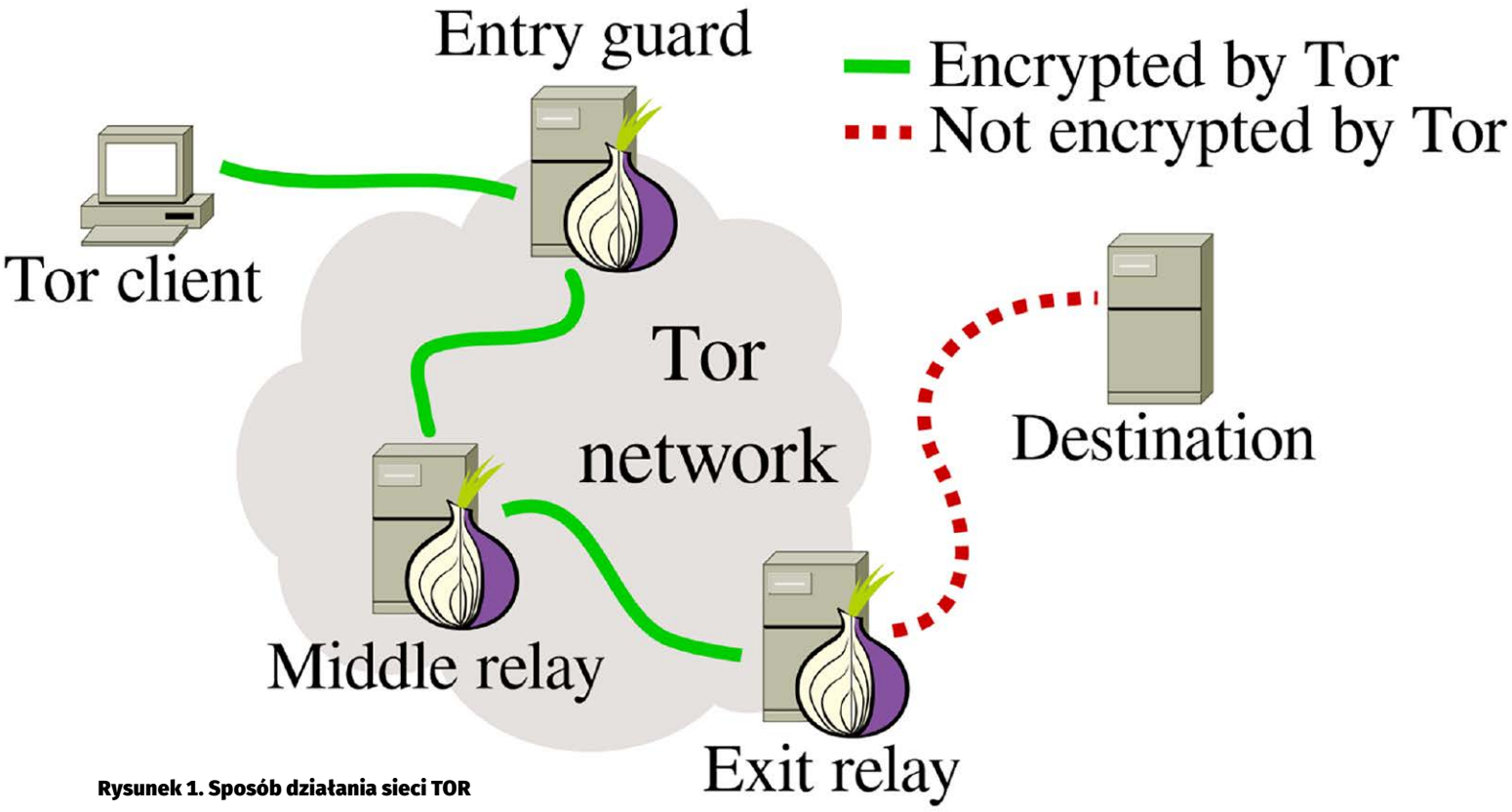


Rysunek 1. Sposób działania sieci TOR

Raspberry TOR Router

Korzystanie z sieci TOR nie jest specjalnie trudne, ale wymaga nieco zachodu. W praktyce każde urządzenie, na którym chcemy puścić ruch sieciowy poprzez szereg anonimowych routerów musi mieć zainstalowane odpowiednie oprogramowanie. Da się to jednak obejść w całkiem łatwy sposób – wystarczy tylko posiadać taki komputerek, jak Raspberry Pi.

Coraz częściej słyszy się o problemie prywatności w sieci. Temat ten nabiera na znaczeniu, gdyż już bardzo istotna część naszego życia wiąże się bezpośrednio z korzystaniem z Internetu, a nowoczesne algorytmy pozwalają korporacjom sprawnie analizować i przewidywać nasze zachowania. Jednym ze sposobów na ukrycie przed osobami postronnymi tego, co robimy w sieci jest wykorzystanie protokołu TOR (The Onion Router). Nazwa nie jest przypadkowa i dobrze obrazuje sposób działania tego protokołu. Bazuje on na tzw. trasowaniu cebulowym, a więc na wielokrotnym szyfrowaniu ruchu w taki sposób, by zmylić ewentualnych „podsłuchujących". Na sieć TOR składa się szereg routerów – najczęściej komputerów, które altruistycznie nastawieni użytkownicy udostępniają, by inni mogli swobodnie korzystać z protokołu. Poszczególne routery szyfrują i deszyfrują przenoszone przez siebie pakiety w taki sposób, że osobie postronnej bardzo trudno odgadnąć, skąd i dokąd przesyłane są dane. I o ile TOR nie jest nie do rozgryzienia, to aby prześledzić trasę pakietów trzeba poświęcić dosyć duże środki, w związku z czym w większości przypadków staje się to zwyczajnie nieopłacalne dla ewentualnego „szpiega".

Pudełko z TORem

Skoro zastosowanie protokołu TOR praktycznie nic nie kosztuje, a próba jego pokonania jest dosyć droga, to dlaczego nie wszyscy z niego korzystają? Są dwa podstawowe problemy. Po pierwsze TOR znacznie spowalnia działanie sieci – szczególnie tych szybkich. Przesyłanie pakietów przez wiele routerów wydłuża ich trasę, znacząco zwiększając pingi i istotnie zmniejszając przepustowość. Drugą trudnością jest instalacja TORa. O ile na komputerze PC nie jest to bardzo skomplikowane, to jednak nie każdy będzie skłonny modyfikować ustawienia systemowe, aby przekierowywać ruch przez TORa. Dotyczy to szczególnie użytkowników mało zaawansowanych, dla których przestawianie się pomiędzy normalną, szybką siecią, a TORem będzie po prostu zbyt skomplikowane. W końcu nie na każdym urządzeniu da się zainstalować odpowiednie oprogramowanie. Czasem użytkownik chcący skorzystać z TORa nie ma uprawnień administratora do komputera, którego używa, a czasem na daną platformę nie ma gotowych aplikacji, które w prosty sposób mogłyby wymusić komunikację „cebulową". Warto przy okazji wspomnieć, że dotyczy to m.in. zdecydowanej większości urządzeń IoT, które mają jedynie proste interfejsy komunikacyjne i zbyt

Fotografia 2. Raspberry TOR Router w obudowie, ale bez połączonych elementów

małą moc obliczeniową (oraz pamięć) by zajmować się złożonym szyfrowaniem.

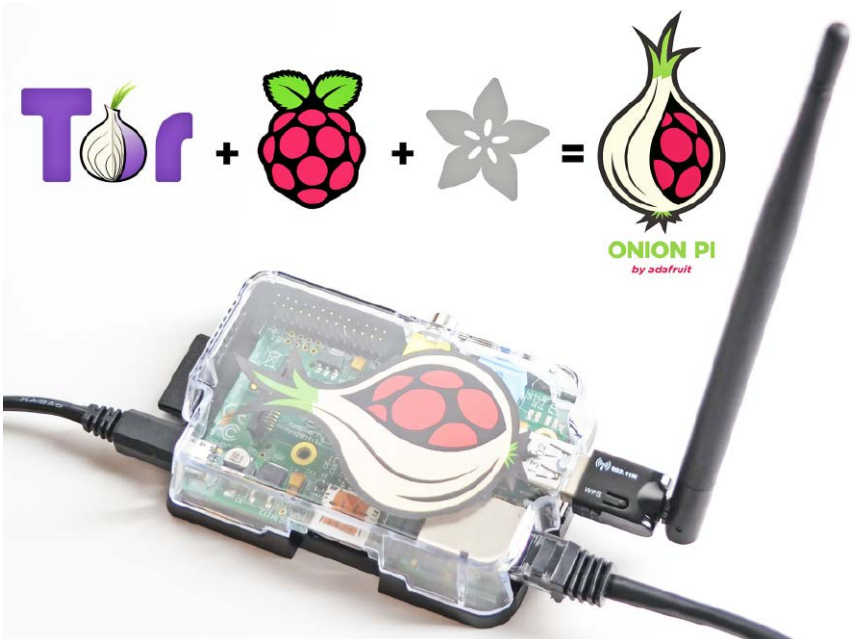
Rozwiązaniem drugiego z tych problemów jest wykorzystanie routera sieci TOR, który automatycznie kieruje cały ruch poprzez serwery rozmieszczone w różnych miejscach świata. Na rynku znaleźć można gotowe routery tego typu w cenie od ok. 60 do 120 USD, różniące się między sobą szczegółami. Jednakże praktycznie taki sam, a ponadto wyposażony w Wi-Fi, możemy samodzielnie wykonać w kilkanaście minut, wykorzystując do tego, choćby stary Raspberry Pi. Kompletny projekt tego typu został opublikowany w serwisie Hackaday.io przez użytkownika o loginie Thomas (***https://goo.gl/VZs1fF***), który wykorzystał narzędzia opracowane przez firmę Adafruit. My publikujemy spolszczoną i zaktualizowaną wersję tego projektu, odtworzoną na Raspberry PI 1 Model B+, ale z najnowszym oprogramowaniem Raspbian Jessie Lite.

Podzespoły

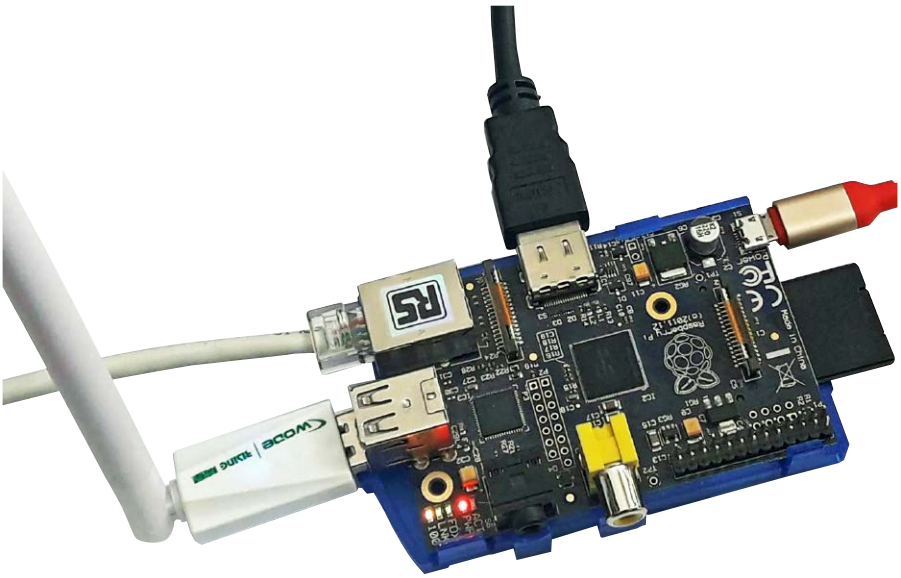
Do wykonania własnego Raspberry TOR Routera nie trzeba wiele. Wystarczy stary Raspberry Pi, 4-gigabajtowa karta pamięci SD, zasilacz USB z kablem microUSB i bezprzewodowa karta sieciowa. W naszym przypadku wykorzystaliśmy niedrogą (wartą ok. 4 USD) kartę 802.11n 150 Mb/s z interfejsem USB 2.0, opartą na chipsecie Realtek RTL8188. Potrzebny jest też przewód do podłączenia Raspberry Pi do sieci przewodowej. Karty Wi-Fi nie należy początkowo podłączać do urządzenia.

Instalacja

Przygotowanie Raspberry TOR Routera można podzielić na trzy etapy. Pierwszy to zapisanie obrazu Raspbiana na kartę pamięci i konfiguracja systemu. Konieczne jest rozszerzenie partycji do maksimum, choć aktualna wersja Raspbiana, którą pobraliśmy (Jessie Lite 2017-04-10) sama wykonała tę operację po pierwszym uruchomieniu systemu. Sensowne będzie też zmienienie hasła (w końcu chcemy uzyskać bezpieczny router sieciowy). Można również ustawić komputer, by automatycznie logował się na konto użytkownika po uruchomieniu (Boot Options –> Console Autologin) i skonfigurować dane związane z lokalizacją: strefę czasową, rodzaj klawiatury, język i sposób prezentacji niektórych informacji. Dzięki temu będzie wygodniej wprowadzało się wszelkie kolejne polecenia. Można też zmniejszyć ilość pamięci przydzielonej na kartę graficzną, gdyż ta w ogóle nie będzie wykorzystywana, poza prezentacją treści w trybie tekstowym. Wystarczy ustawić „Memory Split" w dziale „Advanced Options" na 16 MB.



Fotografia 3. Raspberry PI jako hotspot TOR, przygotowany przez firmę Adafruit



Fotografia 4. Nasz Raspberry TOR Router bez górnej części obudowy

Na koniec dobrze będzie zaktualizować oprogramowanie. W tym celu należy wykonać kolejno polecenia"

```
sudo apt-get update
sudo apt-get upgrade
```

Router Wi-Fi

Drugim krokiem będzie instalacja sterowników do karty sieciowej Wi-Fi i przełączenie jej w tryb pracy punktu dostępowego. Można w tym celu skorzystać z gotowego skryptu, który automatyzuje bardzo wiele ustawień. Można go pobrać poleceniem **`wget https://cdn.hackaday.io/files/4223180676832/pifi.sh`**. My przygotowaliśmy jego spolszczoną, zaktualizowaną wersję pod adresem <**`XXXXXXXXXXXX`**>. Po pobraniu pliku należy nadać mu uprawnienia do wykonywania **`chmod +x pifi.sh`** i można go uruchomić **`sudo ./pifi.sh`**. Skrypt zapyta kolejno o kilka parametrów i przygotuje pliki konfiguracyjne. Gdy zakończy pracę komputer się sam zrestartuje, po czym należy go bezpiecznie wyłączyć. W tym celu warto użyć polecenia **`sudo shutdown -a now`**

Dopiero wtedy należy podpiąć kartę Wi-Fi do złącza USB i włączyć zasilanie. Po ponownym uruchomieniu, w okolicy pojawi

Listing 1. Skrypt konfigurujący Raspberry Pi do pracy jako punkt dostępowy

```
#!/bin/bash

apt-get update -q -y
apt-get install hostapd
apt-get install isc-dhcp-server

cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.bak
sed -i -e ,s/option domain-name „example.org”/# option domain-name „example.org”/g’ /etc/dhcp/dhcpd.conf
sed -i -e ,s/option domain-name-servers ns1.example.org/# option domain-name-servers ns1.example.org/g’ /etc/dhcp/dhcpd.conf
sed -i -e ,s/#authoritative;/authoritative;/g’ /etc/dhcp/dhcpd.conf
echo -e „subnet 192.168.42.0 netmask 255.255.255.0 {
range 192.168.42.10 192.168.42.50;
option broadcast-address 192.168.42.255;
option routers 192.168.42.1;
default-lease-time 600;
max-lease-time 7200;
option domain-name \042local\042;
option domain-name-servers 8.8.8.8, 8.8.4.4;
}” >> /etc/dhcp/dhcpd.conf
cp /etc/default/isc-dhcp-server /etc/default/isc-dhcp-server.bak
sed -i -e ,s/INTERFACES=””/INTERFACES=”wlan0”/g’ /etc/default/isc-dhcp-server

ifdown wlan0

mv /etc/network/interfaces /etc/network/interfaces.bak
echo „auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0

iface wlan0 inet static
  address 192.168.42.1
  netmask 255.255.255.0
„ > /etc/network/interfaces

ifconfig wlan0 192.168.42.1

echo „$(tput bold ; tput setaf 2)Wprowadź nazwę (SSID) nowej sieci, Wi-Fi. Może mieć od 1 do 32 znaków. Zakończ wprowadzanie klawiszem [EN-
TER]:$(tput sgr0)”
read ssid
echo „$(tput setaf 6)Nazwa sieci została zapisana jako: $(tput bold)$ssid$(tput sgr0 ; tput setaf 6). Możesz edytować plik /etc/hostapd/
hostapd.conf by ją zmienić.$(tput sgr0)”

pwd1=”0”
pwd2=”1”
until [ $pwd1 == $pwd2 ]; do
  echo „$(tput bold ; tput setaf 2)Wprowadź hasło dostępowe do stworzonej sieci w naciśnij [ENTER]:$(tput sgr0)”
  read -s pwd1
  echo „$(tput bold ; tput setaf 2)Dla weryfikacji powtórz wprowadzone przed chwilą hasło i naciśnij [ENTER]:$(tput sgr0)”
  read -s pwd2
done

if [ $pwd1 == $pwd2 ]; then
  echo „$(tput setaf 6)Hasło zostało ustalone. Możesz je zmienićw pliku /etc/hostapd/hostapd.conf.$(tput sgr0)”
fi

echo „interface=wlan0
driver=rtl871xdrv
ssid=$ssid
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=$pwd1
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP” > /etc/hostapd/hostapd.conf

cp /etc/default/hostapd /etc/default/hostapd.bak
sed -i -e ,s/#DAEMON_CONF=””/DAEMON_CONF=”\/etc\/hostapd\/hostapd.conf”/g’ /etc/default/hostapd
cp /etc/sysctl.conf /etc/sysctl.bak
echo „net.ipv4.ip_forward=1” >> /etc/sysctl.conf
echo „up iptables-restore < /etc/iptables.ipv4.nat” >> /etc/network/interfaces
sh -c „echo 1 > /proc/sys/net/ipv4/ip_forward”
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -i eth0 -o wlan0 -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
sh -c „iptables-save > /etc/iptables.ipv4.nat”

wget http://www.adafruit.com/downloads/adafruit_hostapd.zip
unzip adafruit_hostapd.zip
mv /usr/sbin/hostapd /usr/sbin/hostapd.ORIG
mv hostapd /usr/sbin
chmod 755 /usr/sbin/hostapd
rm adafruit_hostapd.zip

service hostapd start
service isc-dhcp-server start
service hostapd status
hostapd_result=$?

#if [ $hostapd_result == 3 ]; then
#  echo „ERROR: hostapd start failed.”
#  exit 1
#fi

service isc-dhcp-server status
dhcp_result=$?

#if [ $dhcp_result == 3 ]; then
#  echo „ERROR: ISC DHCP server failed to start.”
#  exit 1
#fi

update-rc.d hostapd enable
update-rc.d isc-dhcp-server enable
mv /usr/share/dbus-1/system-services/fi.epitest.hostap.WPASupplicant.service ~/

reboot

exit 0
```

się widoczna nowa sieć bezprzewodowa o nazwie SSID takiej, jaka została podana w trakcie konfiguracji.

Co robi skrypt pifi.sh? Główna część przygotowanej przez nas polskiej wersji została umieszczona na **listingu 1**. Skrypt kolejno aktualizuje informacje o dostępnych pakietach, po czym instaluje dwa programy. Hostapd, który odpowiada właśnie za pracę karty sieciowej jako bezprzewodowego punktu dostępowego oraz isc-dhcp-server, który po prostu jest serwerem DHCP. Dzięki temu punkt dostępowy może automatycznie przydzielać adresy IP poszczególnym podłączanym urządzeniom.

Kolejne linijki odpowiadają za konfigurację serwera DHCP. Będzie on rozdawał IP w zakresie od 192.168.42.10 do 192.168.42.50. Sam punkt dostępowy przyjmuje adres 192.168.42.1 i będzie podawać się za bramkę sieci bezprzewodowej. Serwery DNS zostały ustawione na googlowe 8.8.8.8 i 8.8.4.4, dzięki czemu całość będzie działać w prawie dowolnym miejscu na świecie (pewnie za wyjątkiem Chin). Po tych operacjach rekonfigurowane są interfejsy sieciowe, aż w końcu serwer pyta o nazwę i hasło tworzonego hot spotu.

Kolejny krok to konfiguracja programu hostapd. Jak widać, wykorzystywany sterownik nie odpowiada bezpośrednio temu, który mamy w naszej karcie sieciowej, ale to nie szkodzi gdyż w praktyce ustawienia te działały z kilkoma kartami, jakie podłączyliśmy. Następne kilka liniej odpowiada za wymuszenie uruchamiania programu hotspotu zaraz po włączeniu systemu oraz za konfigurację firewalla, tak by odpowiednio kierował ruchem. Wykorzystywany jest do tego popularny program iptables. Gdy ta operacja się zakończy, skrypt pobiera spakowane dane zaktualizowanego programu hostapd z serwera Adafruit, dzięki czemu mamy pewność, że będzie on poprawnie działał z Raspberry Pi. Po instalacji skrypt restartuje usługi i sprawdza, czy się uruchomiły. Jeśli wszystko jest ok, włącza ich uruchamianie przy starcie i restartuje komputer.

Router TOR

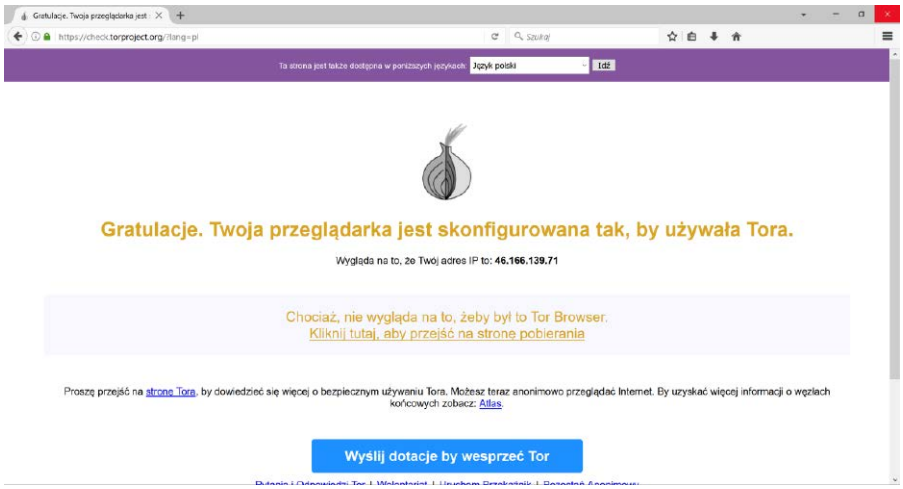
Po pomyślnym restarcie należy uruchomić drugi skrypt. Można go pobrać za pomocą komendy **`wget https://cdn.hackaday.io/files/4223180676832/tor.sh`**, a uruchomić nadając odpowiednie uprawnienia **`chmod +x tor.sh`** i wykonując **`sudo ./tor.sh`**.

Skrócona, polska wersja drugiego skryptu została zamieszczona na **listingu 2**. Skrypt kolejno pobiera i instaluje oprogramowanie TOR, po czym je konfiguruje. Wprowadza przy tym ustawienia zgodne z określonymi w trakcie pracy pierwszego ze skryptów. Następne kroki dotyczą ustawień firewalla. Warto zauważyć, że wyjątkowo traktowany

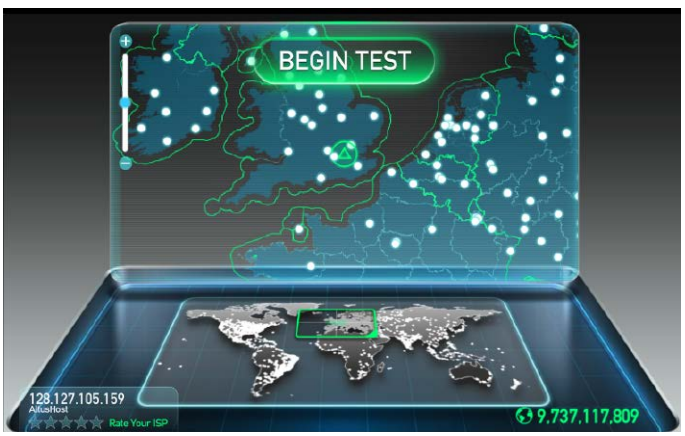
Fotografia 5. Różne karty sieciowe na USB, z którymi sprawdzaliśmy działanie routera.

Listing 2. Skrypt konfigurujący Raspberry Pi do pracy w sieci TOR

```
#!/bin/bash

apt-get install tor -y

cp /etc/tor/torrc /etc/tor/torrc.bak
echo „Log notice file /var/log/tor/notices.log
VirtualAddrNetwork 10.192.0.0/10
AutomapHostsSuffixes .onion,.exit
AutomapHostsOnResolve 1
TransPort 9040
TransListenAddress 192.168.42.1
DNSPort 53
DNSListenAddress 192.168.42.1” >> /etc/tor/torrc

iptables -F
iptables -t nat -F
iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 22 -j REDIRECT --to-ports 22
iptables -t nat -A PREROUTING -i wlan0 -p udp --dport 53 -j REDIRECT --to-ports 53
iptables -t nat -A PREROUTING -i wlan0 -p tcp --syn -j REDIRECT --to-ports 9040
sh -c „iptables-save > /etc/iptables.ipv4.nat”

touch /var/log/tor/notices.log
chown debian-tor /var/log/tor/notices.log
chmod 644 /var/log/tor/notices.log

service tor start
update-rc.d tor enable
reboot

exit 0
```



Rysunek 6. Test pokazuje, że łączymy się przez sieć TOR



Rysunek 7. Próba przetestowania szybkości połączenia internetowego

Rysunek 8. Wynik testów szybkości połączenia internetowego

jest port 22, który pozwoli zdalnie rekonfigurować Raspberry Pi przez SSH. Konfigurowane jest też rejestrowanie zdarzeń w logach i automatyczne uruchamianie TORa, po czym system jest restartowany.

Podsumowanie

Z naszego doświadczenia wynika, że potrzeba kilku minut od uruchomienia routera do momentu, aż zacznie on w sprawnie przekazywać ruch przez sieć TOR. Niemniej wszelkie podłączone komputery były widziane, jakby przeglądały Internet zza TORa. Można to sprawdzić na kilka sposobów. Pierwszy to wejście na stronę ***https://check.torproject.org***. Rezultat podczas korzystania z przygotowanego przez nas hotspotu widać na rysunku. Można też pokusić się o przetestowanie szybkości łącza. W naszym przypadku serwis Speedtest.net uznał, że łączymy się z Wielkiej Brytanii (**rysunek 7**), a uzyskane wyniki pomiarów (**rysunek 8**) zaprezentowano nam z francuskojęzycznymi reklamami.

Marcin Karbowniczek, EP