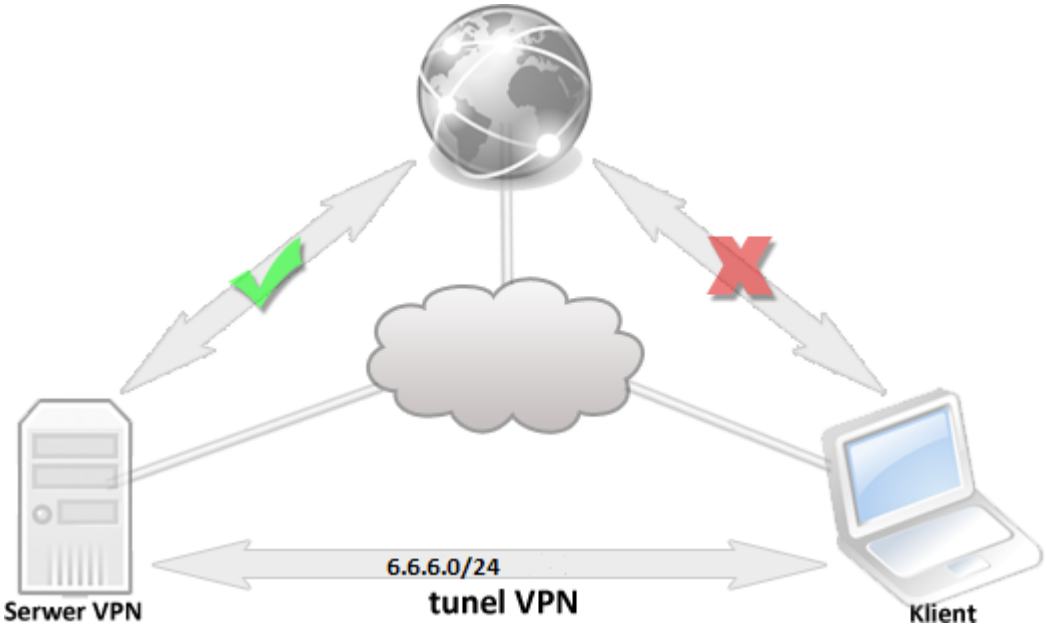
Jakiś czas temu pisałem o ataku typu MITM (Man In The Middle). Dziś zajmiemy się jednym z sposobów zapobiegania takim atakom. Co prawda nie będziemy zapobiegać na poziomie sieci ale na poziomie naszego komputera. Czym jest to rozwiązanie? To VPN (Virtual Private Network).

Tradycyjnie zajrzyjmy do wikipedii:

**VPN** (ang. Virtual Private Network, Wirtualna Sieć Prywatna) – tunel, przez który płynie ruch w ramach sieci prywatnej pomiędzy klientami końcowymi za pośrednictwem
publicznej sieci (takiej jak Internet) w taki sposób, że węzły tej sieci są przezroczyste dla
przesyłanych w ten sposób pakietów. Można opcjonalnie kompresować lub szyfrować przesyłane
dane w celu zapewnienia lepszej jakości lub większego poziomu bezpieczeństwa.

Tłumacząc to na język prostych ludzi:
VPN – tunel łaczący nasz komputer z serwerem VPN. Ruch w tunelu dla przełączników sieciowych jak i wszelakich intruzów chcących nas podsłuchiwać jest niewidoczny. Rozwiązanie to, często jest wykorzystywane w dużych firmach, gdzie pracownicy pracują zdalnie, dużo podróżują i zmuszeni są używać niezaufanego łącza.

A jak wygląda to w następujący sposób:




Co będzie nam potrzebne? Na początek potrzebujemy serwera który będzie naszym serwerem VPN. Ja do tego celu wykorzystuje serwer hostowany w rootbox.pl. Już opcja „Small" pozwala nam na wygodne korzystanie z własnego vpn'a. Będziemy tez potrzebować dwóch maszyn wirtualnych. Dlaczego dwóch? Już tłumacze. Dziś nie tylko skonfiguruje połączenie VPN'a, ale też i przeprowadzę ponownie atak MITM na maszynę podłączoną tunelem, aby wykazać, brak możliwości podsłuchania.

Pierwszym krokiem jest stworzenie naszego serwera. Ja osobiście wybieram zawsze Debiana 64, jest to wersja, z którą najlepiej mi się pracuje. Po chwili od kliknięcie „utwórz" w panelu rootbox w swoim mailu można znaleźć wiadomość z danymi dostępowymi. Czas więc zacząć instalacje.

Spis treści


- Konfiguracja serwera.
- Kilka praktycznych uwag dotyczących połączeń VPN.
- Konfiguracja klienta – Windows
- Konfiguracja klienta – Linux (Debian, gnome)
- Pro-tip na zwiększenie bezpieczeństwa.
- Atak MITM.

# Konfiguracja serwera.

Logujemy się do naszego vps'a i zaczynamy od aktualizacji. Jest to dobra praktyka, nigdy nie wiemy kiedy serwerownia pre konfigurowała nasz obraz.

```
root@vpn-blog:~# apt-get update && apt-get upgrade
```

Teraz instalujemy Openvpn.

```
root@vpn-blog:~# apt-get install openvpn
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  liblzo2-2 libpkcs11-helper1
Suggested packages:
  openssl resolvconf
The following NEW packages will be installed:
  liblzo2-2 libpkcs11-helper1 openvpn
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 621 kB of archives.
After this operation, 1,489 kB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Klikamy „Y" i instalujemy nie tylko Openvpn ale i wymagane dodatkowo pakiety.

Teraz gdy mamy już zainstalowany openvpn, zajmiemy się generowaniem kluczy. Przedtem, jeżeli nie mamy zainstalowanego instalujemy openssl.

```
root@vpn-blog:~/easy-rsa# apt-get install openssl
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
 ca-certificates
The following NEW packages will be installed:
 openssl
```

```
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 701 kB of archives.
After this operation, 1,108 kB of additional disk space will be used.
Get:1 http://debian.wdc.pl/security/ wheezy/updates/main openssl amd64
1.0.1e-2+deb7u14 [701 kB]
Fetched 701 kB in 0s (9,615 kB/s)
Selecting previously unselected package openssl.
(Reading database ... 20844 files and directories currently
installed.)
Unpacking openssl (from .../openssl_1.0.1e-2+deb7u14_amd64.deb) ...
Processing triggers for man-db ...
Setting up openssl (1.0.1e-2+deb7u14) ...
```

Następnym krokiem będzie:

```
root@vpn-blog:~# pwd
/root
root@vpn-blog:~# cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0/
./easy-rsa
root@vpn-blog:~# cd easy-rsa/
```

Dla bezpieczeństwa narzędzia do generowania kluczy warto trzymać w katalogu /root. Dlaczego? Bo tylko root będzie miał do niego dostęp.

Następnie edytujemy plik vars

```
# Increase this to 2048 if you
# are paranoid.  This will slow
# down TLS negotiation performance
# as well as the one-time DH parms
# generation process.
export KEY_SIZE=2048

# In how many days should the root CA key expire?
export CA_EXPIRE=3650

# In how many days should certificates expire?
export KEY_EXPIRE=3650
```

```
# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY="PL"
export KEY_PROVINCE="MAZOWIECKIE"
export KEY_CITY="Warszawa"
export KEY_ORG="S.M.S Security"
export KEY_EMAIL="pht@s-m-s.org.pl"
export KEY_EMAIL=pht@s-m-s.org.pl
export KEY_CN=VPN-Blog
export KEY_NAME=VPN-Blog
export KEY_OU=VPN-Blog
export PKCS11_MODULE_PATH=changeme
export PKCS11_PIN=1234
```

Ustawiamy długość klucza na 2048, bo jesteśmy paranoikami oraz edytujemy ostatnie linie, po to by nie musieć uzupełniać ich za każdym razem jak będziemy generować kolejne klucze dla klientów.

Teraz należy wczytać zmiany, które wprowadziliśmy.

```
root@vpn-blog:~/easy-rsa# source ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /root/easy-
rsa/keys
```

Dla pewności używamy ./clean-all by upewnić się, że nie mamy jakiś przypadkowo wygenerowanych kluczy.

Tworzymy tzw. klucz urzędu certyfikacji CA oraz generujemy klucz Diffiego-Hellmana:

```
root@vpn-blog:~/easy-rsa# ./build-ca
Generating a 2048 bit RSA private key
..........................+++
.....................................................+++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will
be incorporated
into your certificate request.
```

```
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [PL]:PL
State or Province Name (full name) [MAZOWIECKIE]:
Locality Name (eg, city) [Warszawa]:
Organization Name (eg, company) [S.M.S Security]:
Organizational Unit Name (eg, section) [VPN-Blog]:
Common Name (eg, your name or your server's hostname) [VPN-Blog]:
Name [VPN-Blog]:
Email Address [pht@s-m-s.org.pl]:
root@vpn-blog:~/easy-rsa# ./build-dh
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.....................................+.......+..........+.........
[...]
.....................+.....................................++*++*
```

Generowanie kluczy serwera:

```
root@vpn-blog:~/easy-rsa# ./build-key-server klucz-serwera
Generating a 2048 bit RSA private key
.........................................+++
.....+++
writing new private key to 'klucz-serwera.key'
-----
You are about to be asked to enter information that will
be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
```

```
Country Name (2 letter code) [PL]:
State or Province Name (full name) [MAZOWIECKIE]:
Locality Name (eg, city) [Warszawa]:
Organization Name (eg, company) [S.M.S Security]:
Organizational Unit Name (eg, section) [VPN-Blog]:
Common Name (eg, your name or your server's hostname) [klucz-serwera]:
Name [VPN-Blog]:
Email Address [pht@s-m-s.org.pl]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /root/easy-rsa/openssl-1.0.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'PL'
stateOrProvinceName :PRINTABLE:'MAZOWIECKIE'
localityName :PRINTABLE:'Warszawa'
organizationName :PRINTABLE:'S.M.S Security'
organizationalUnitName:PRINTABLE:'VPN-Blog'
commonName :PRINTABLE:'klucz-serwera'
name :PRINTABLE:'VPN-Blog'
emailAddress :IA5STRING:'pht@s-m-s.org.pl'
Certificate is to be certified until Jan 14 21:05:35 2025 GMT (3650
days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Następnie generujemy klucze dla naszego klienta.

```
root@vpn-blog:~/easy-rsa# ./build-key client
Generating a 2048 bit RSA private key
.........................................+++
```

```
.........................................+++
writing new private key to 'client.key'
-----
You are about to be asked to enter information that will
be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [PL]:
State or Province Name (full name) [MAZOWIECKIE]:
Locality Name (eg, city) [Warszawa]:
Organization Name (eg, company) [S.M.S Security]:
Organizational Unit Name (eg, section) [VPN-Blog]:
Common Name (eg, your name or your server's hostname) [client]:
Name [VPN-Blog]:
Email Address [pht@s-m-s.org.pl]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /root/easy-rsa/openssl-1.0.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'PL'
stateOrProvinceName   :PRINTABLE:'MAZOWIECKIE'
localityName          :PRINTABLE:'Warszawa'
organizationName      :PRINTABLE:'S.M.S Security'
organizationalUnitName:PRINTABLE:'VPN-Blog'
commonName            :PRINTABLE:'client'
name                  :PRINTABLE:'VPN-Blog'
emailAddress          :IA5STRING:'pht@s-m-s.org.pl'
Certificate is to be certified until Jan 14 21:20:16 2025 GMT (3650
```

```
days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
root@vpn-blog:~/
```

Kiedy już wygenerowaliśmy na tą chwile potrzebne nam klucze. Warto dać upust swojej paranoicznej naturze i zwiększyć stopień bezpieczeństwa. Utworzymy dodatkowy klucz dla mechanizmu *tls-auth,* którego głównym zadaniem jest odrzucanie na wczesnym etapie połączeń od nieautoryzowanych klientów. Dzięki niemu uzyskamy również dodatkową ochronę przed podatnościami typu przepełnienie bufora, występującymi w implementacji protokołu SSL/TLS.

```
root@vpn-blog:~/easy-rsa# openvpn --genkey --secret ta.key
root@vpn-blog:~/easy-rsa# mv ta.key keys/
```

Teraz czas na skopiowanie kluczy w odpowiednie miejsca.

```
cp keys/ca.crt /etc/openvpn/
cp keys/klucz-serwera.crt /etc/openvpn/
cp keys/klucz-serwera.key /etc/openvpn/
cp keys/dh2048.pem /etc/openvpn/dh.pem
cp keys/ta.key /etc/openvpn/
```

Teraz należy stworzyć plik konfiguracyjny.

```
root@vpn-blog:~/easy-rsa# cd /etc/openvpn/
root@vpn-blog:/etc/openvpn# nano openvpn.conf
```

Może on wyglądać tak:

```
port 1194
proto udp
dev tun
ca ca.crt
cert klucz-serwera.crt
key klucz-serwera.key
```

```
dh dh.pem
server 6.6.6.0 255.255.255.0
keepalive 10 120
tls-auth ta.key 0
comp-lzo
user nobody
cipher AES-256-CBC
group nogroup
persist-key
persist-tun
log-append openvpn.log
verb 3
mute 10
```

Następująco uruchamiamy translacje adresów, aby nasz serwer był bramką do internetu.

```
root@vpn-blog:/etc/openvpn# sed -i "/exit 0/d" /etc/rc.local
root@vpn-blog:/etc/openvpn# echo "/sbin/iptables -t nat -A POSTROUTING
-o eth0 -j MASQUERADE" >> /etc/rc.local
root@vpn-blog:/etc/openvpn# echo "exit 0" >> /etc/rc.local
root@vpn-blog:/etc/openvpn# /etc/rc.local
root@vpn-blog:/etc/openvpn# echo "net.ipv4.conf.all.forwarding = 1" >>
/etc/sysctl.conf
root@vpn-blog:/etc/openvpn# sysctl -p
net.ipv4.conf.all.forwarding = 1
```

Dodajemy do pliku konfiguracyjnego

```
push "redirect-gateway def1"
push "dhcp-option DNS 8.8.8.8" #serwery dns google
push "dhcp-option DNS 8.8.4.4"
```

I uruchamiamy usługę.

```
/etc/init.d/openvpn start
```

## Kilka praktycznych uwag dotyczących połączeń VPN.

Jak już pisałem, z VPN'a można korzystać z różnych powodów. W domu, pracy, miejscach publicznych. Powyższa konfiguracja jest dobra gdy korzystamy w domu do łączenia się firmą, do łączenia komputera w firmie do naszej domowej sieci czy też do logowania do banku w publicznych sieciach. Można też wykorzystać tego typu połączenie do zestawiania bezpiecznych połączeń z serwerami backupów, bazodanowymi czy deweloperskimi, które nie mają wyjścia na świat innego niż vpn. Oczywiście przede wszystkim VPN pozwoli nam omijać wszelaką cenzurę. Gdy mówimy o cenzurze w firmie, np na facebooka, należy skonfigurować nasz VPN na port 443, ponieważ pozostałe mogą być wycięte na firewallu. Natomiast jeżeli nasza siedziba firmy jest chroniona poprzez Next Generation Firewall taki jak PaloAlto… Cóż wiele nie zdziałamy, ale o tym w osobnym artykule.

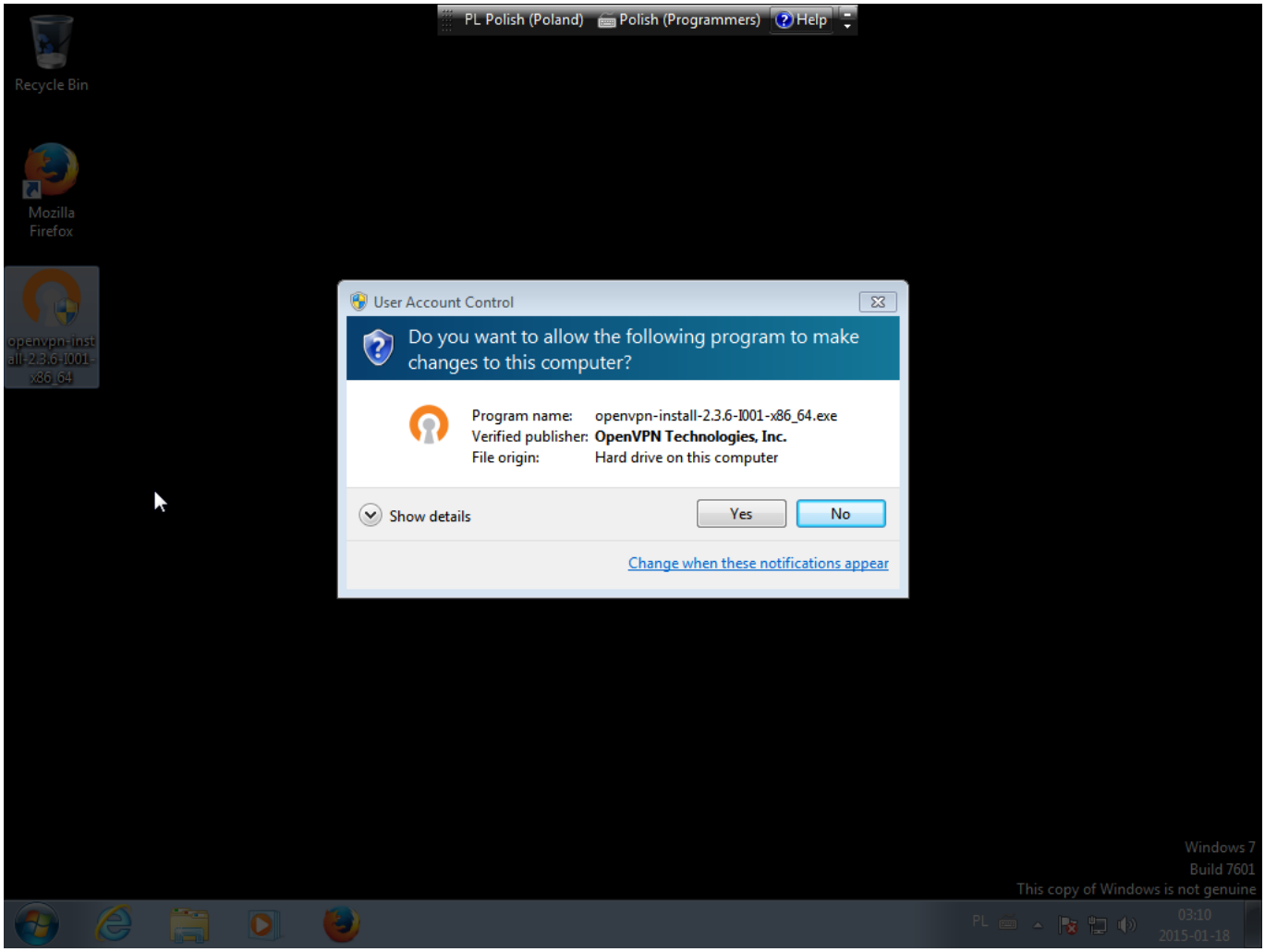
Jeżeli, ktoś chce anonimizować swoją tożsamość a nie ją chronić przed kradzieżą, polecam mu sieć TOR. Można też korzystać z sieci vpn'ów SecurityKiss. Ale jaki sens ma posiadanie vpn'a wspólnego z ludzi, których nawet nie znamy.

## Konfiguracja klienta – Windows

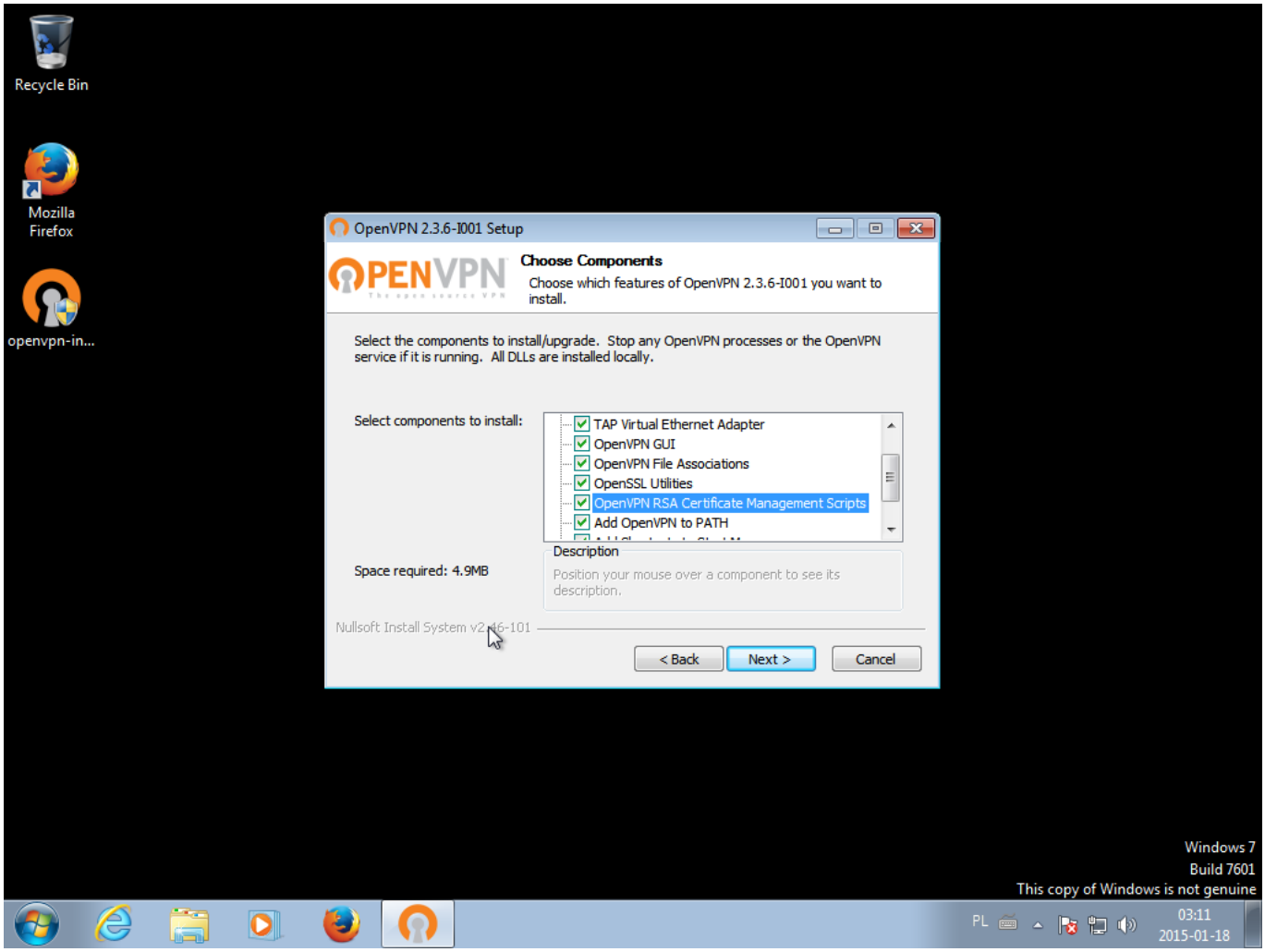
Przyszedł czas na konfiguracje klienta. Zaczniemy od pobrania klienta openvpna z gui, ułatwi to nam prace.

Po pobraniu, zajmijmy się instalacją.

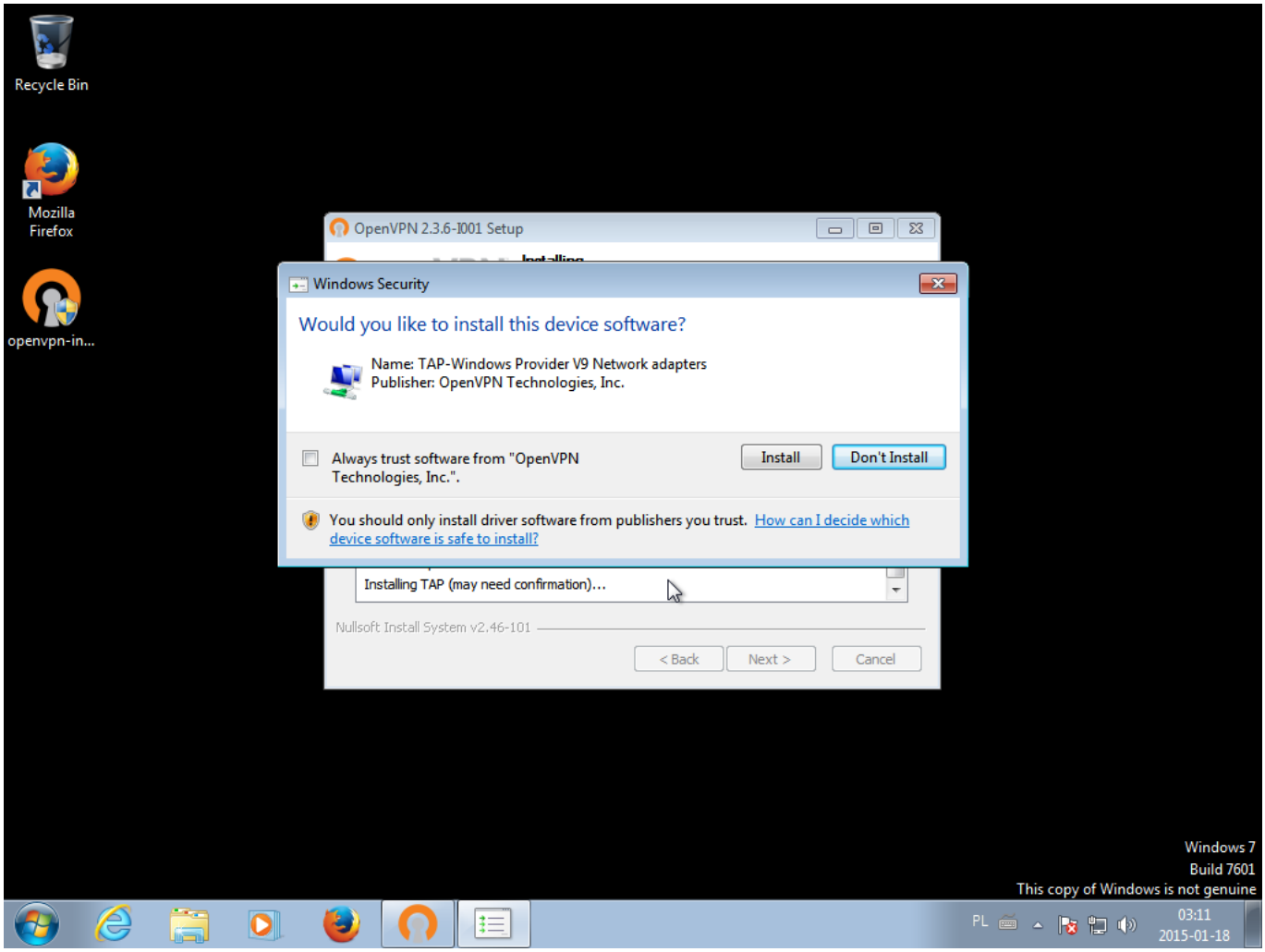
Uwaga, w przypadku windows 7 i wyżej należy instalacje oraz program uruchamiać z prawami administratora.



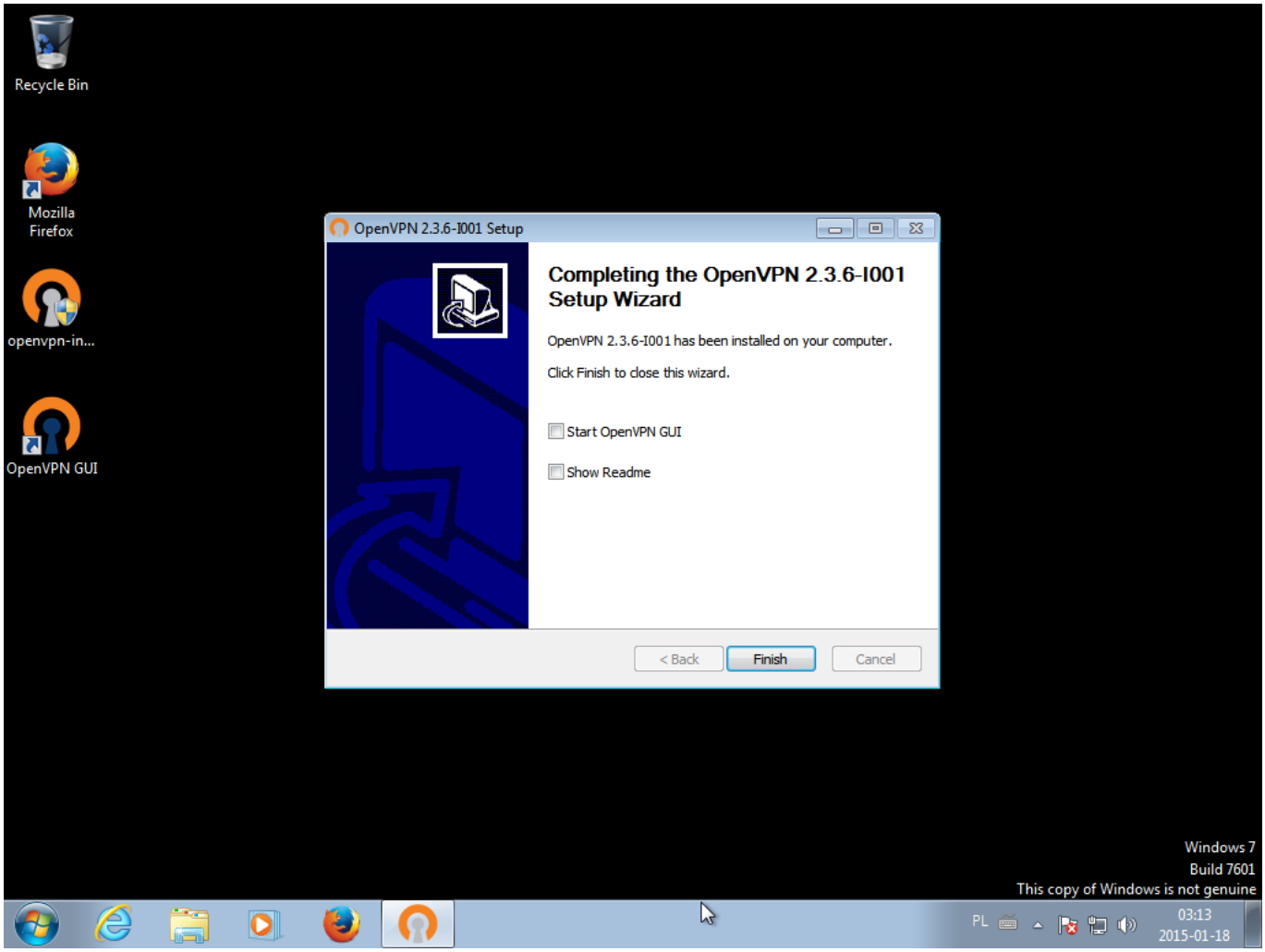

Zaznaczamy wszystkie okienka przy instalacji.



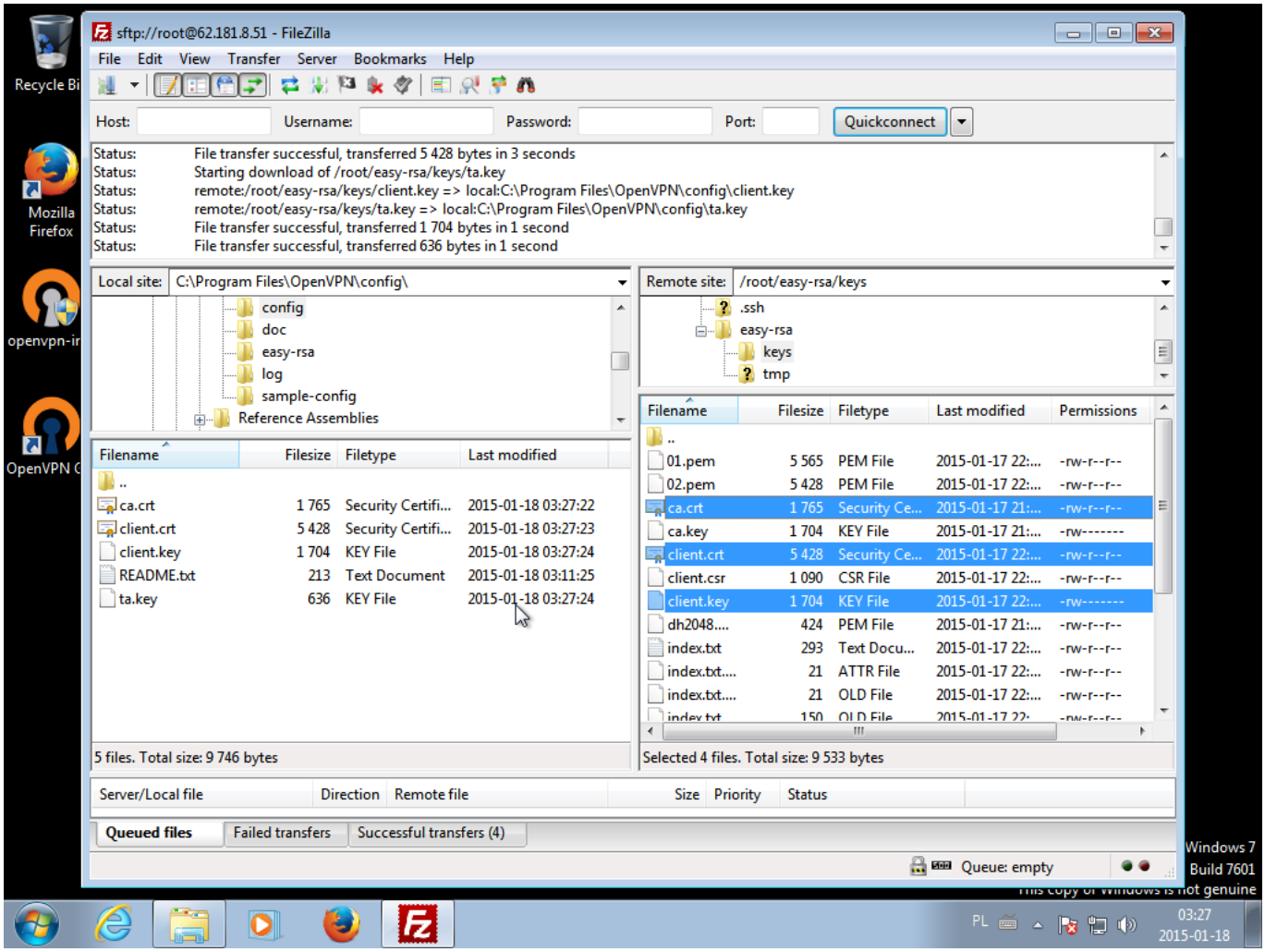

Instalujemy interfejs sieciowy, który będzie interfejsem tunelu.



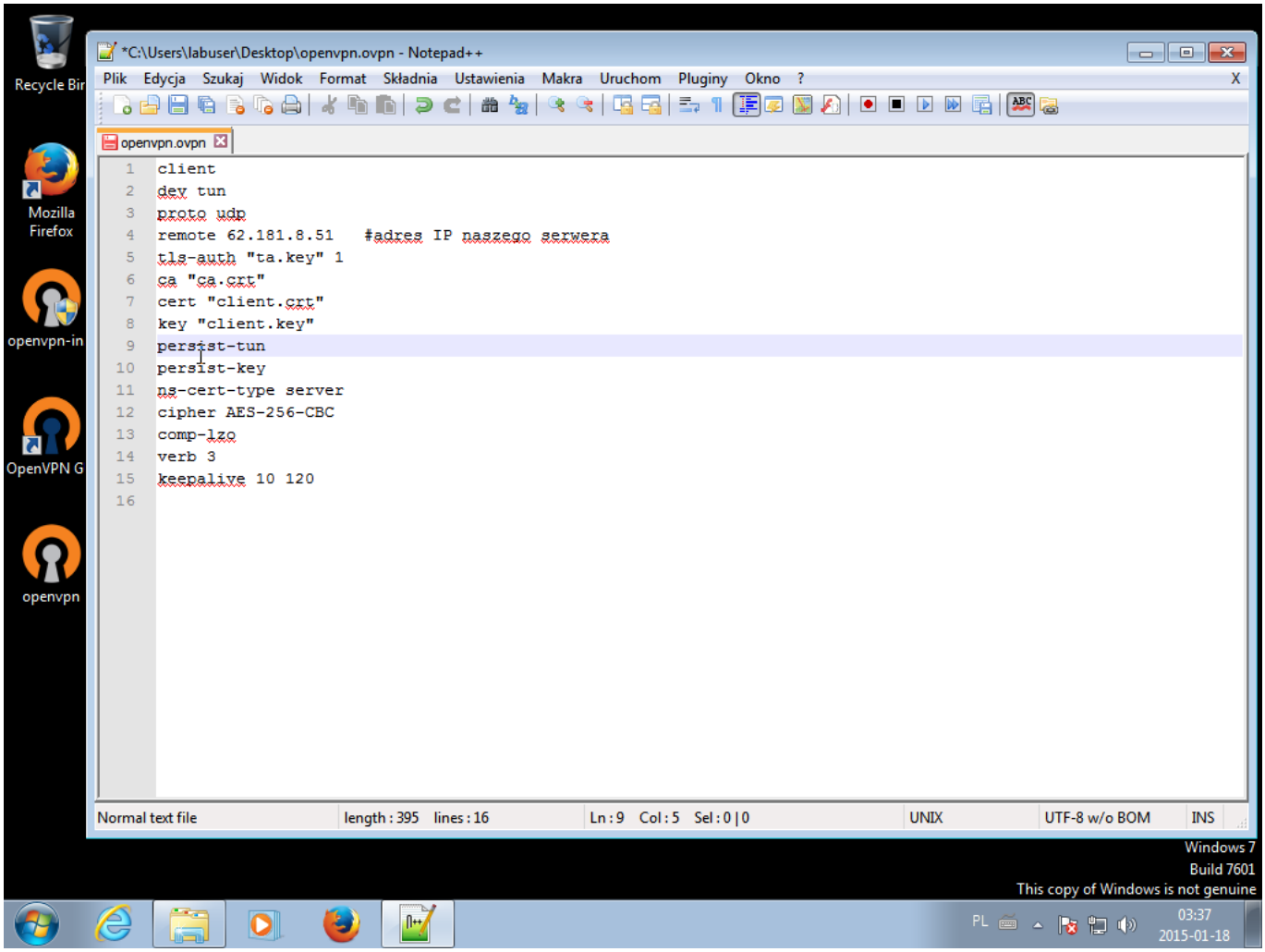

Po instalacji nie uruchamiamy naszego klienta.



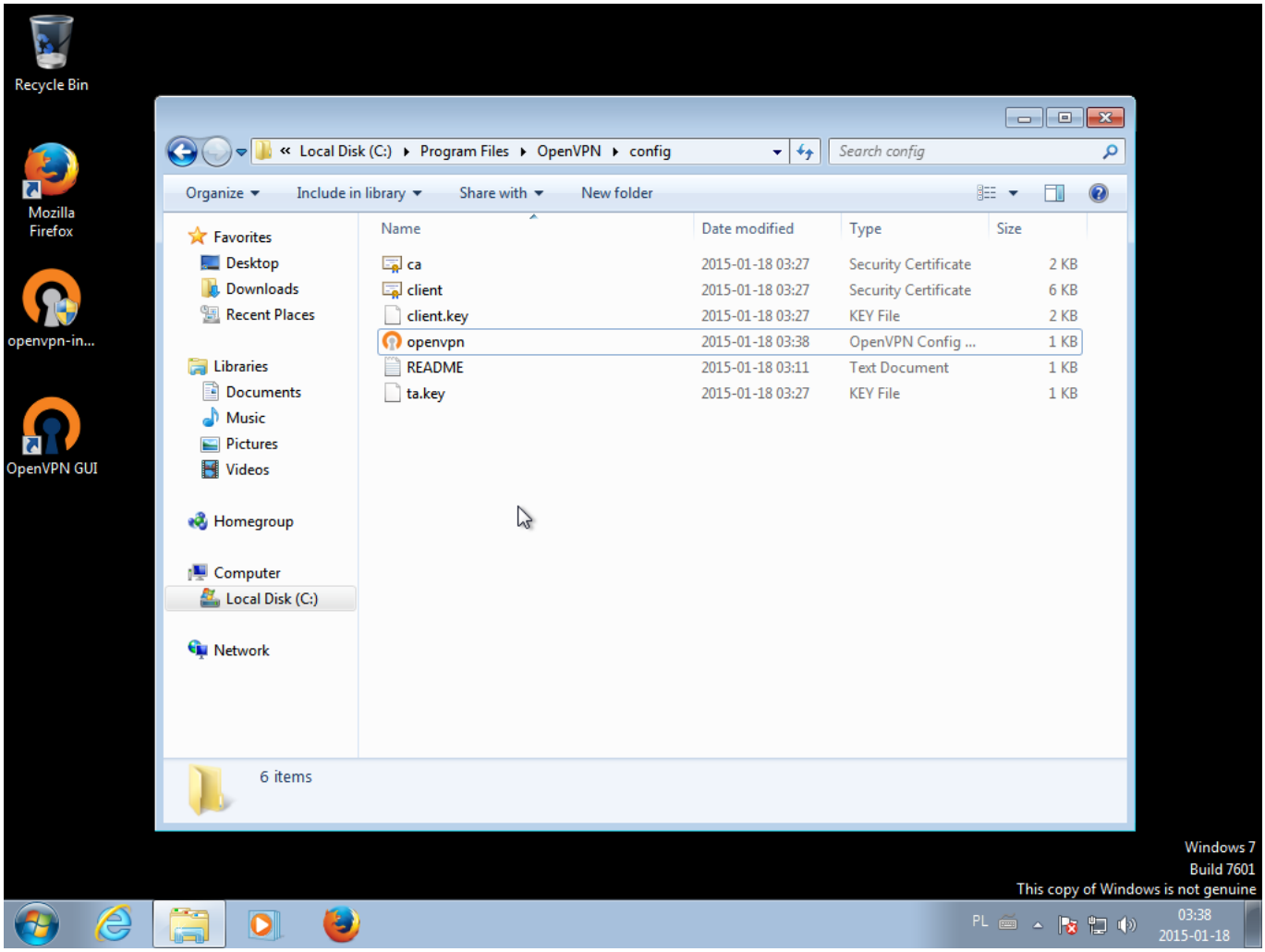

Do katalogu w którym zainstalowaliśmy Openvpn w folderze config wrzucamy ca.crt, client.crt, client.key oraz ta.key.




Przygotowywujemy plik configuracyjny openvpn.

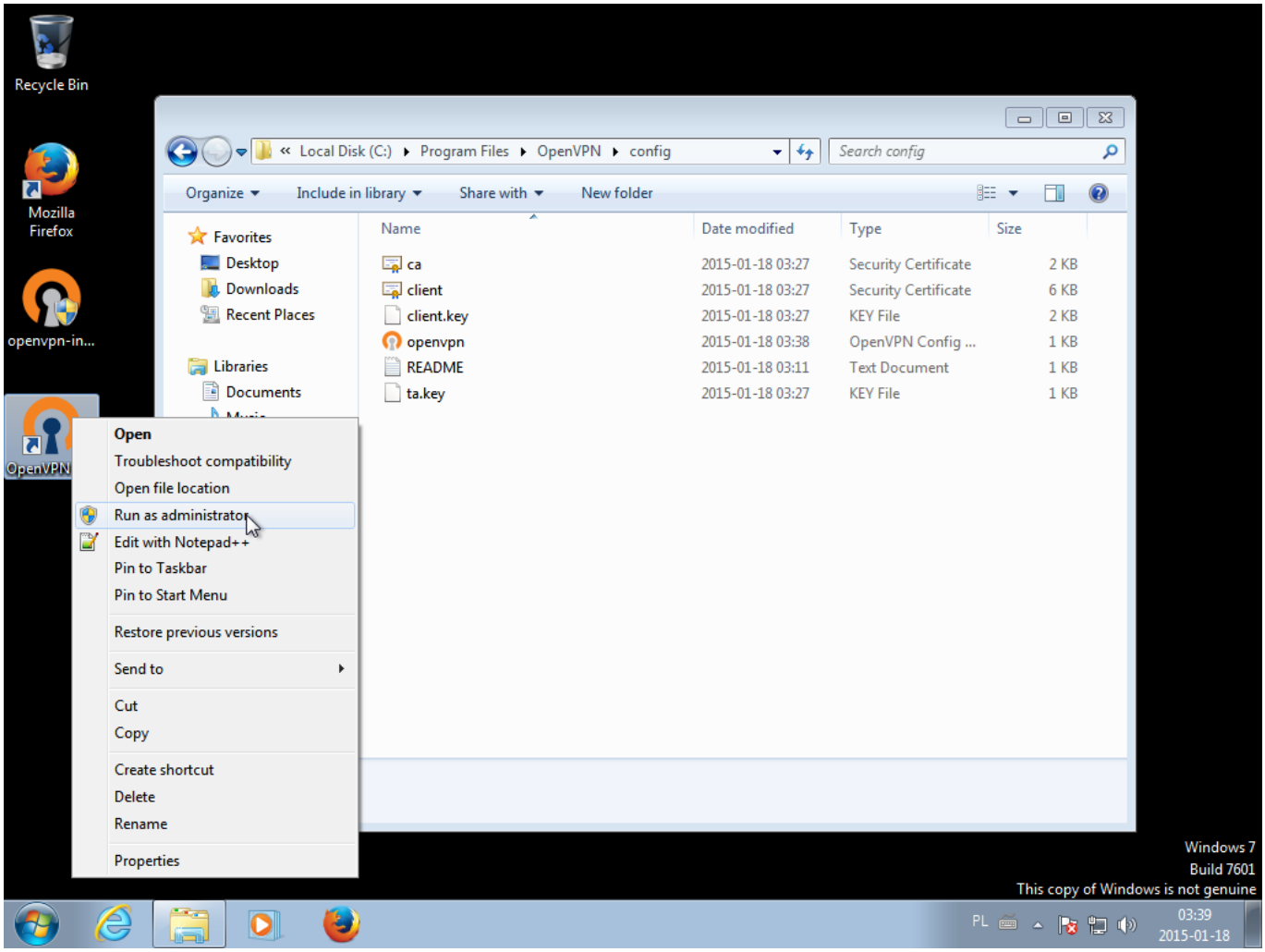



Plik z konfiguracją umieszczamy w katalogu config.

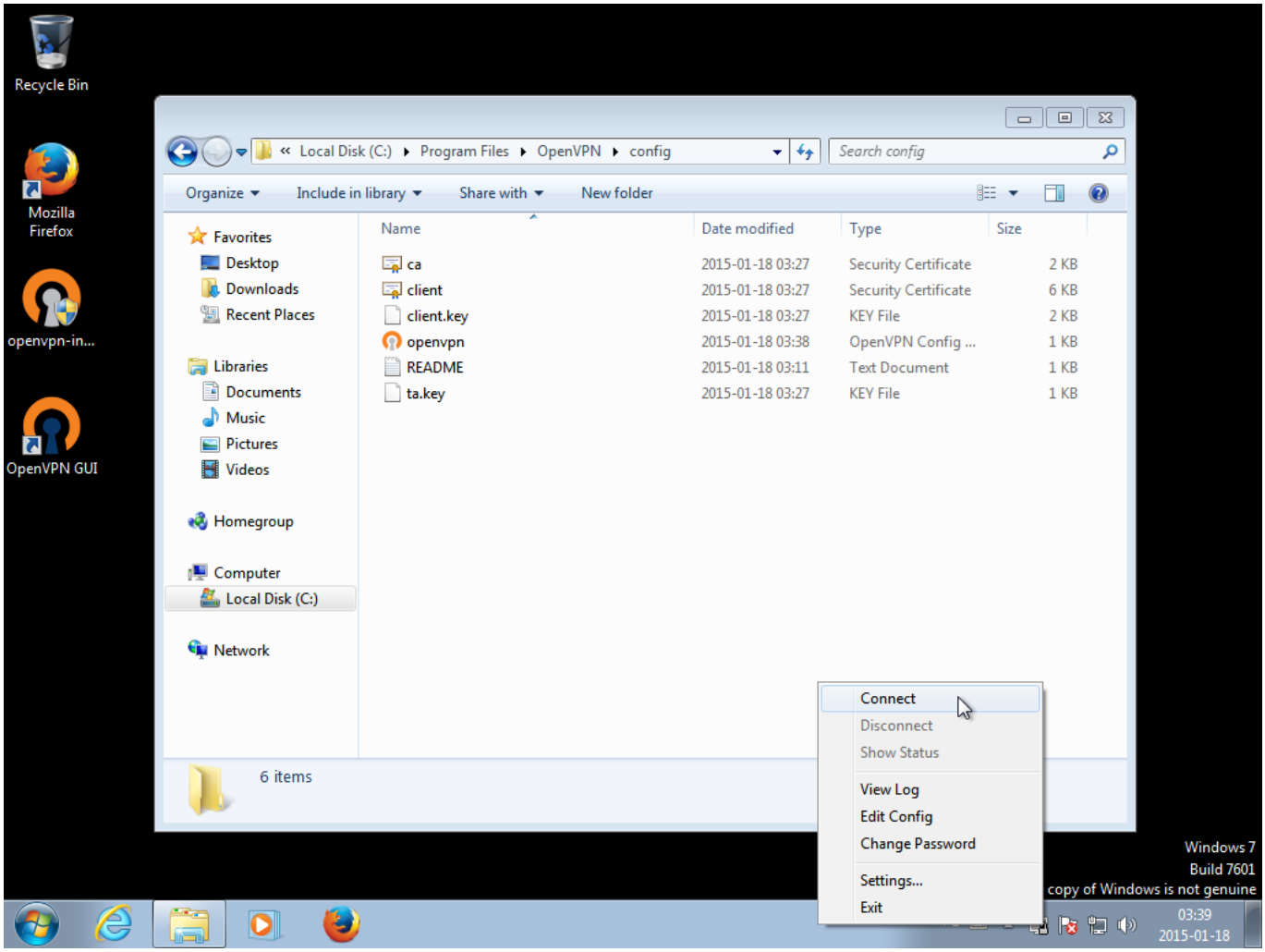



Sprawdźmy teraz nasze ip.

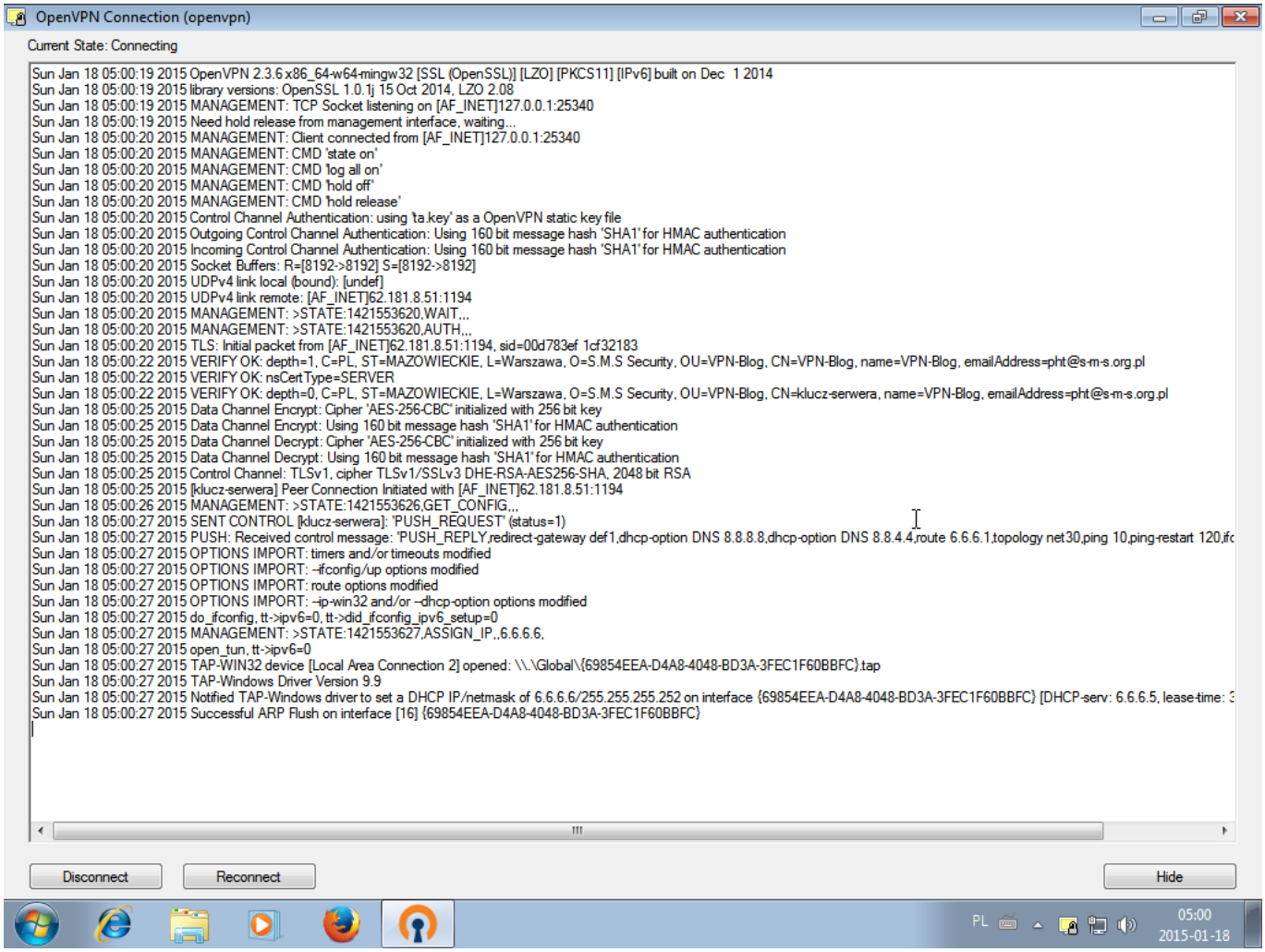

Uruchamiamy klienta openvpn jako administrator.




Na pasku pojawiła się nam ikonka połączenia vpn. Klikamy prawym na „Connect”.




W pojawiającym się oknie ujrzymy logi połączenia i ewentualne informacje o błędach.



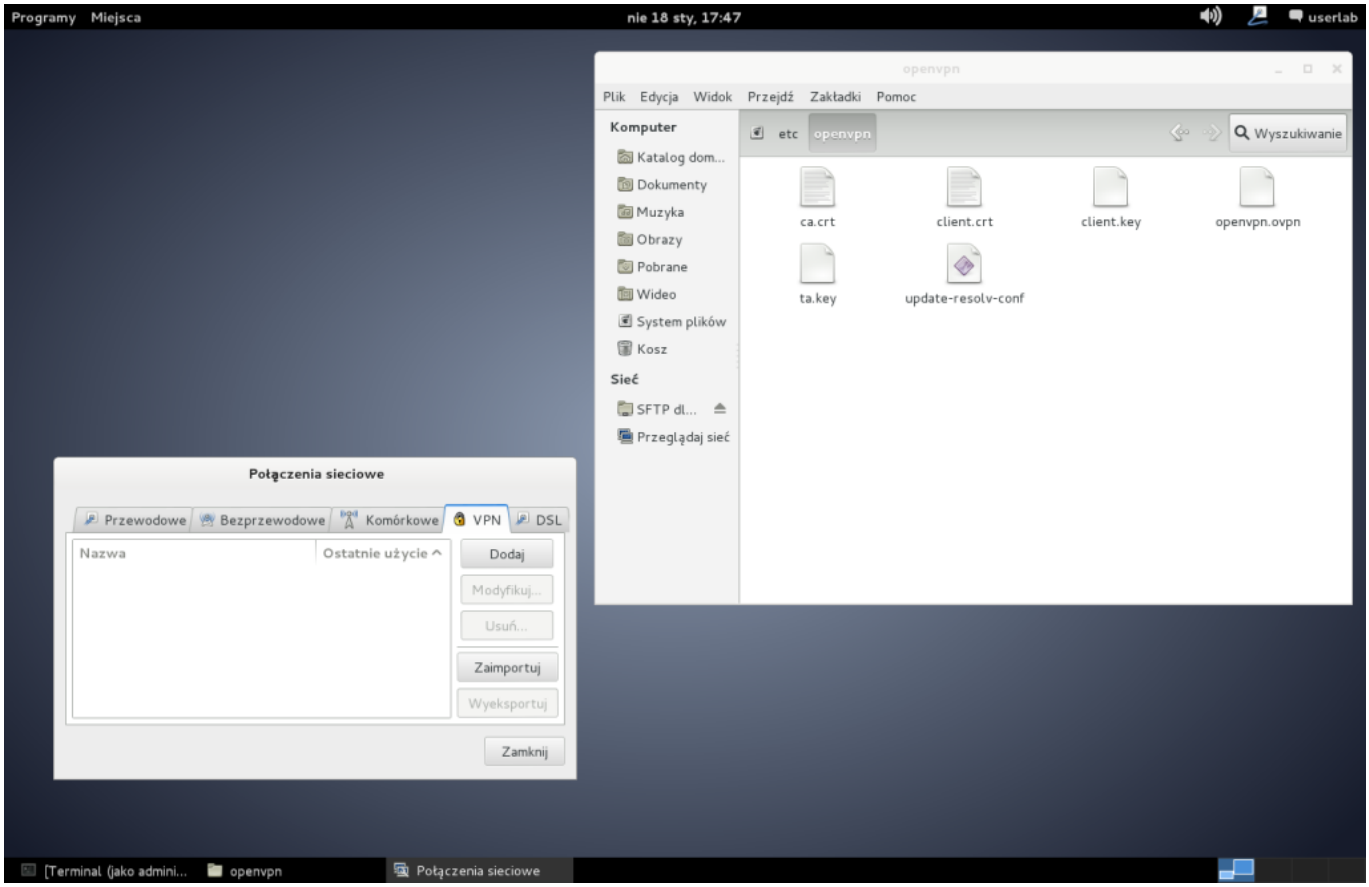

Po udanym połączeniu, sprawdzamy nasze ip. Udało się, teraz nasz ruch jest kierowany przez vpn.


# Konfiguracja klienta – Linux (Debian, gnome)

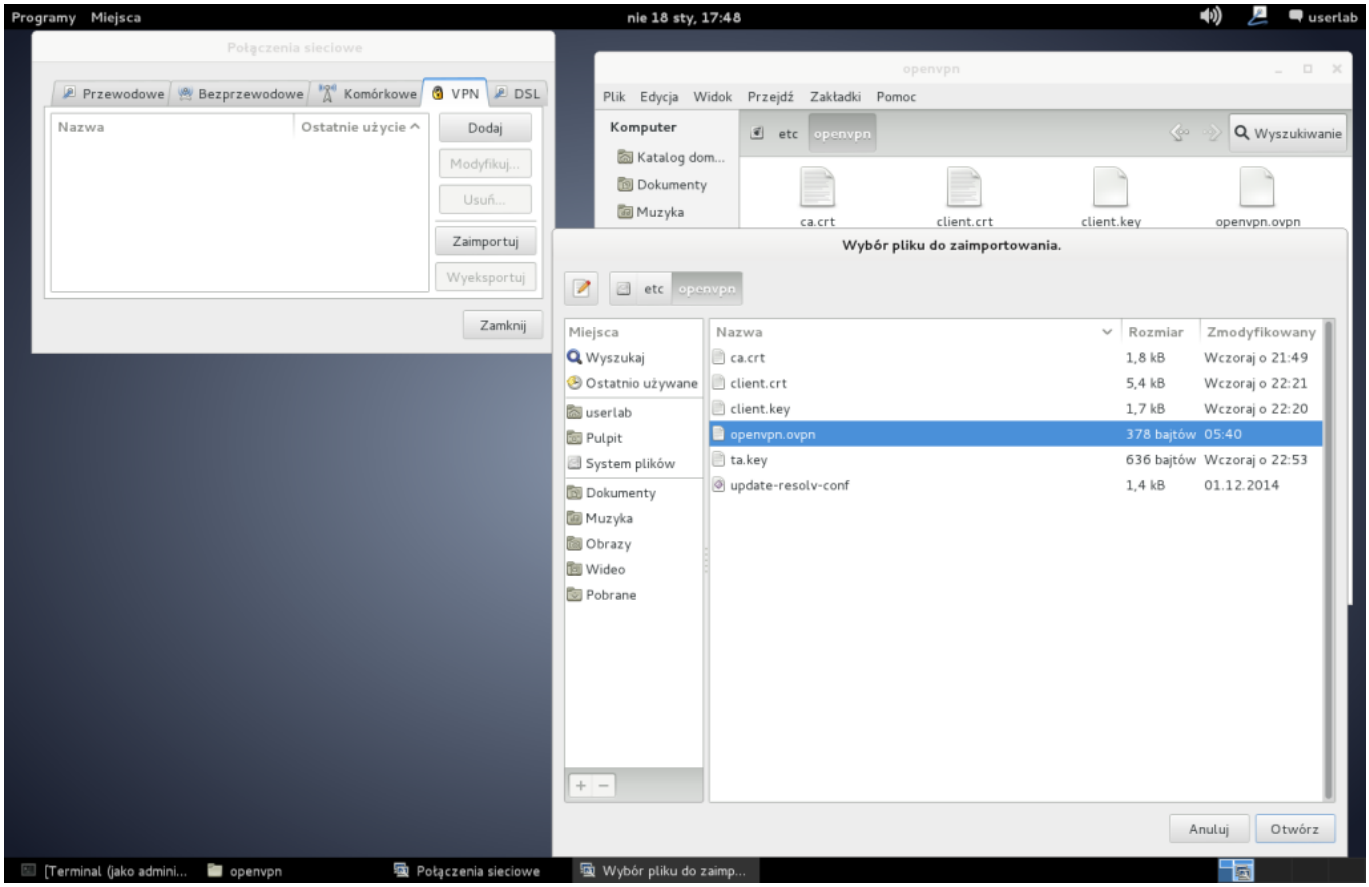
Zacznynamy od instalacji pakietów o ile juz ich nie mamy. Oczywiście

```
apt-get install openvpn  network-manager-openvpn-gnome
```

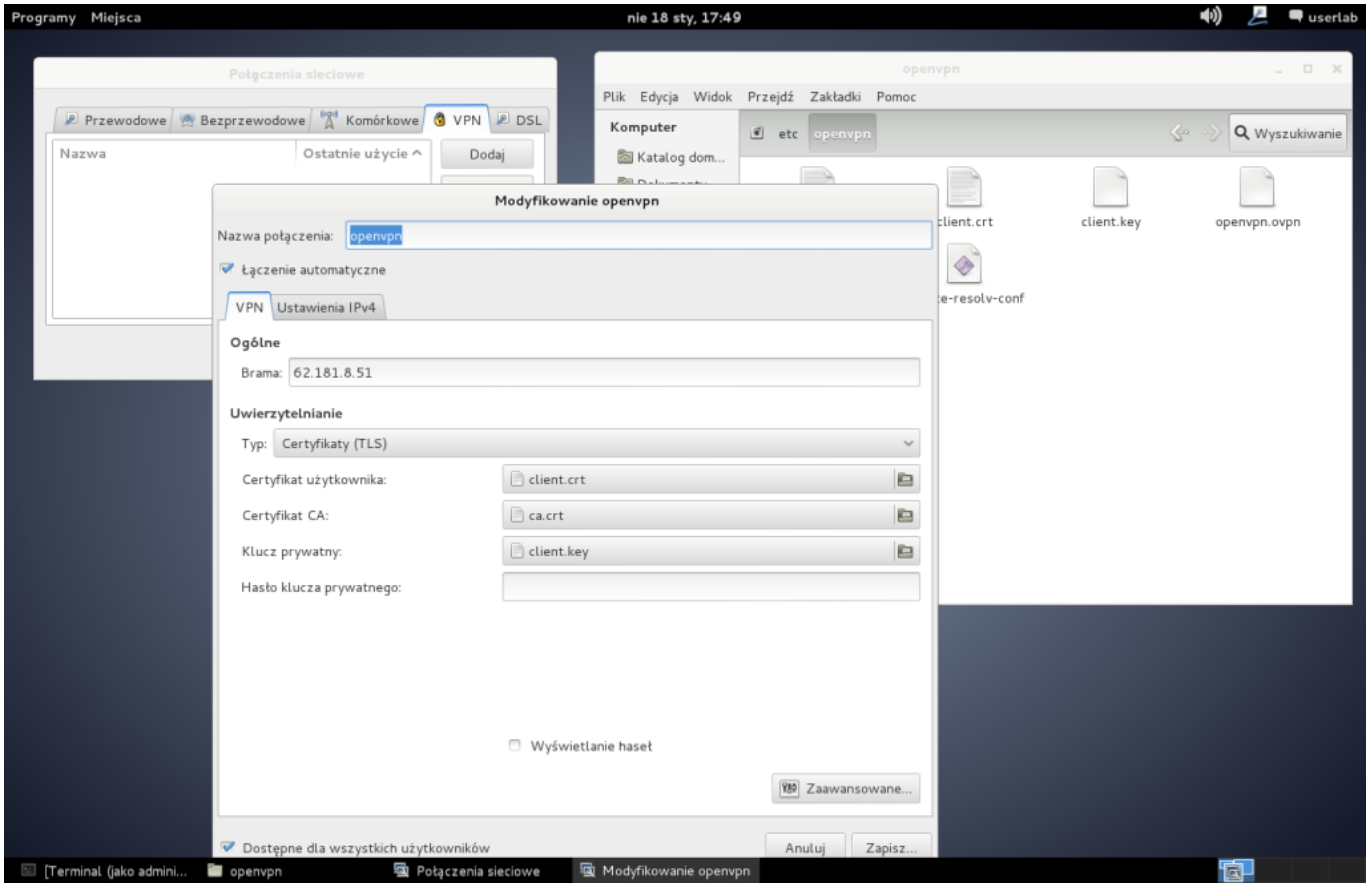
Otwieramy networkmanager'a -> zakładka VPN -> zaimportuj.



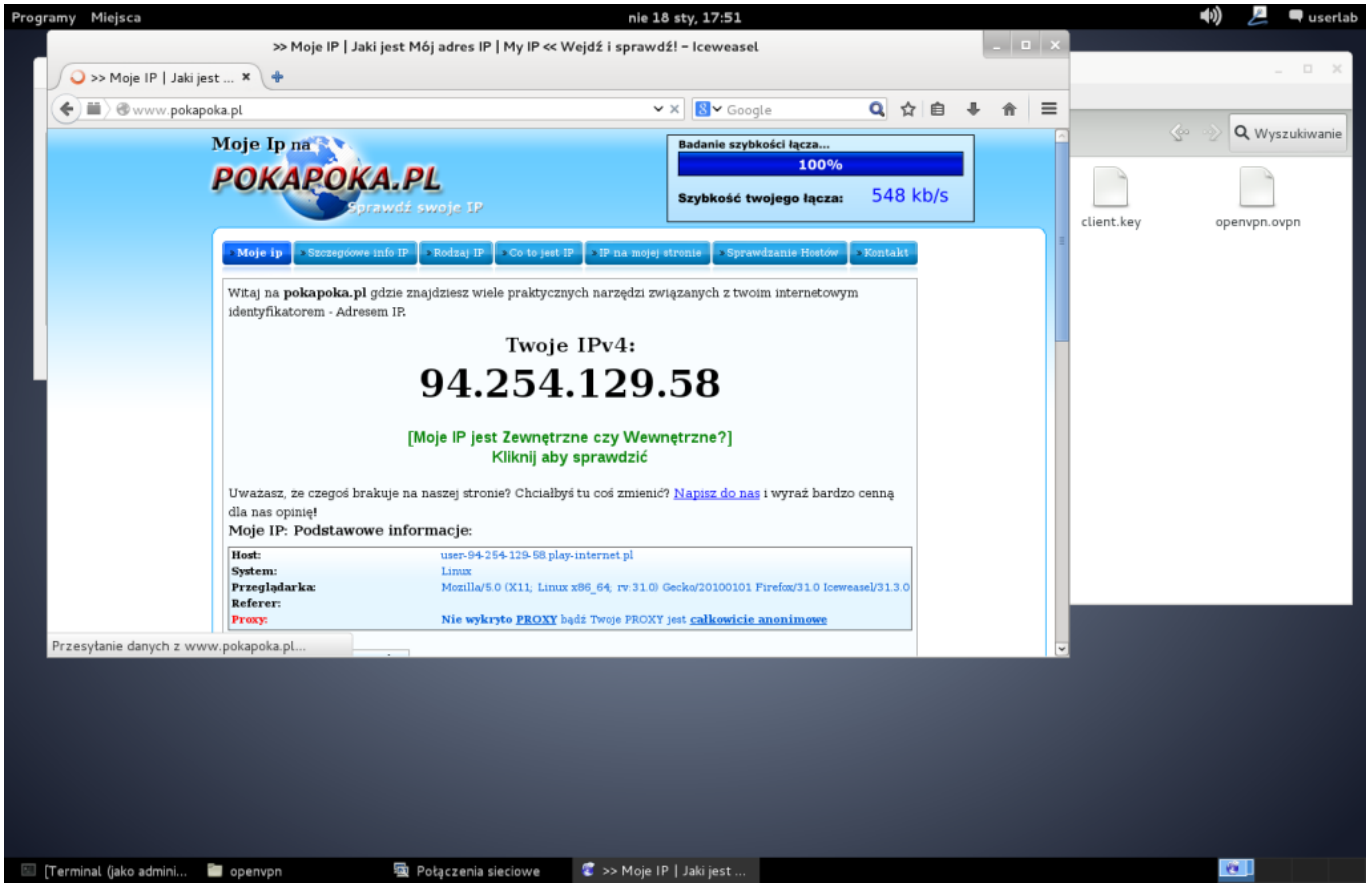

Znajdujemy ca.crt, client.crt, client.key oraz ta.key, które pobraliśmy z serwera vpn.
Wybieramy plik konfiguracji .ovpn



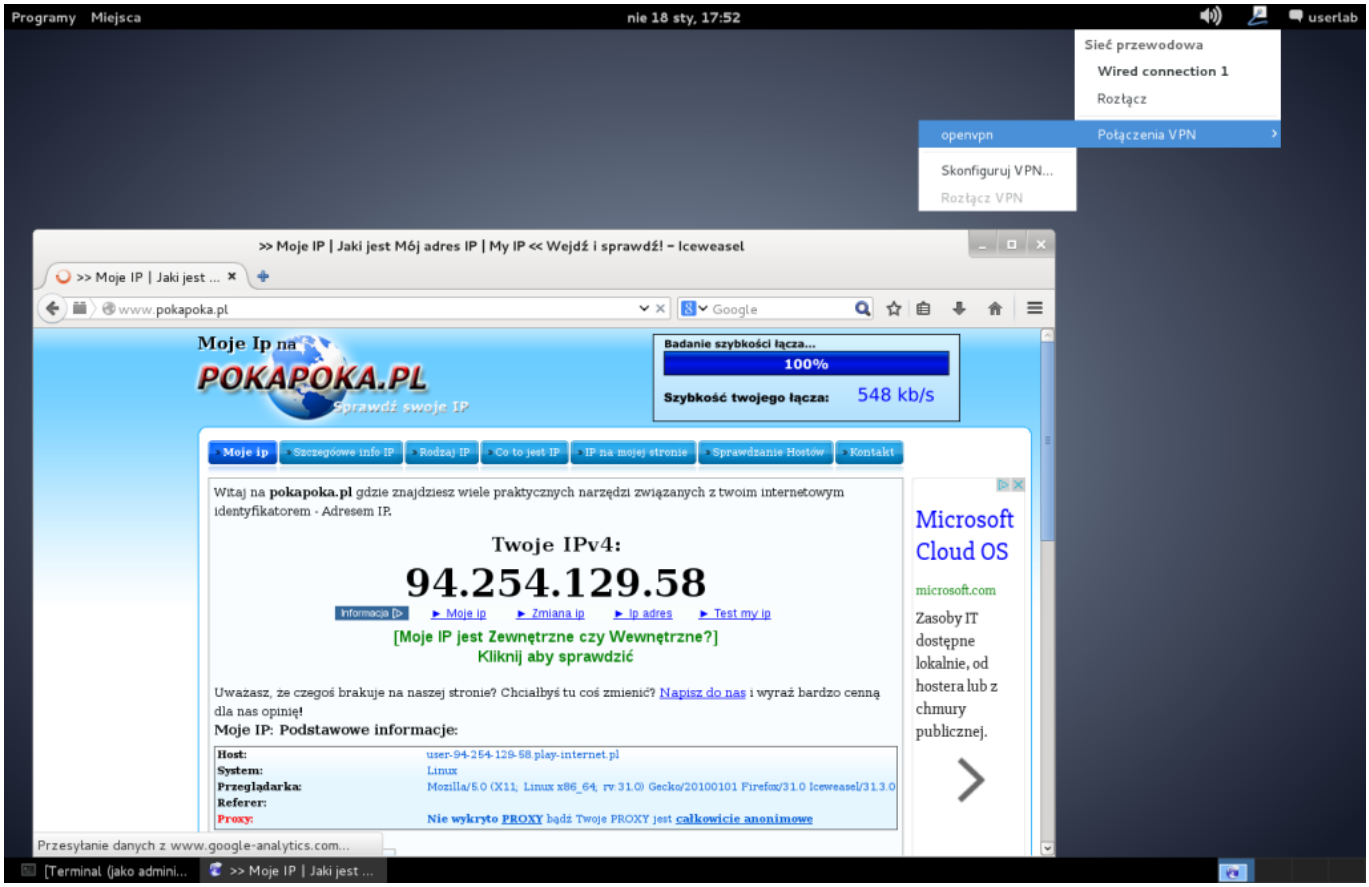

Klikamy zapisz.



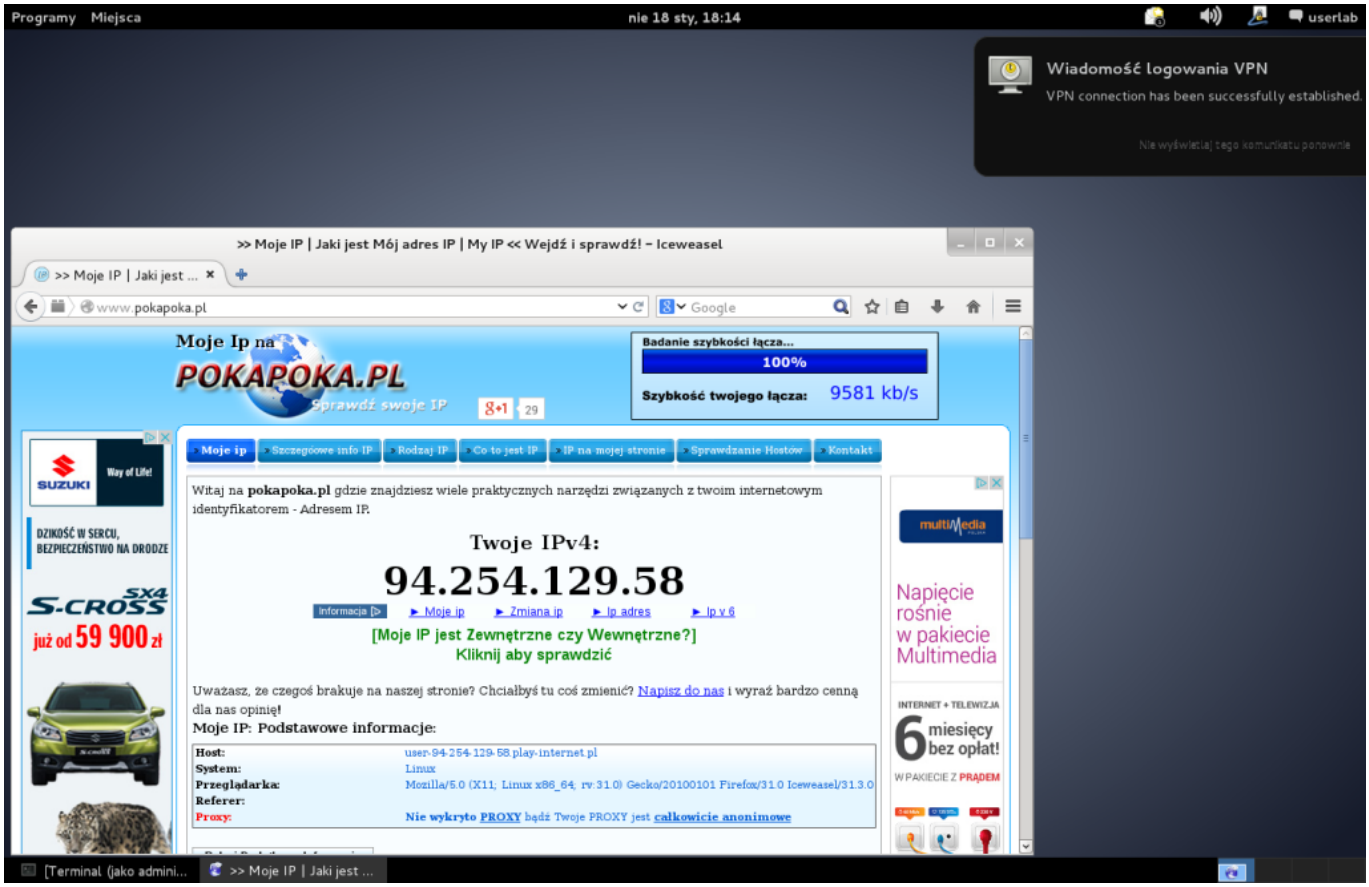

Sprawdźmy teraz nasze IP przed połączeniem się do vpn’a.



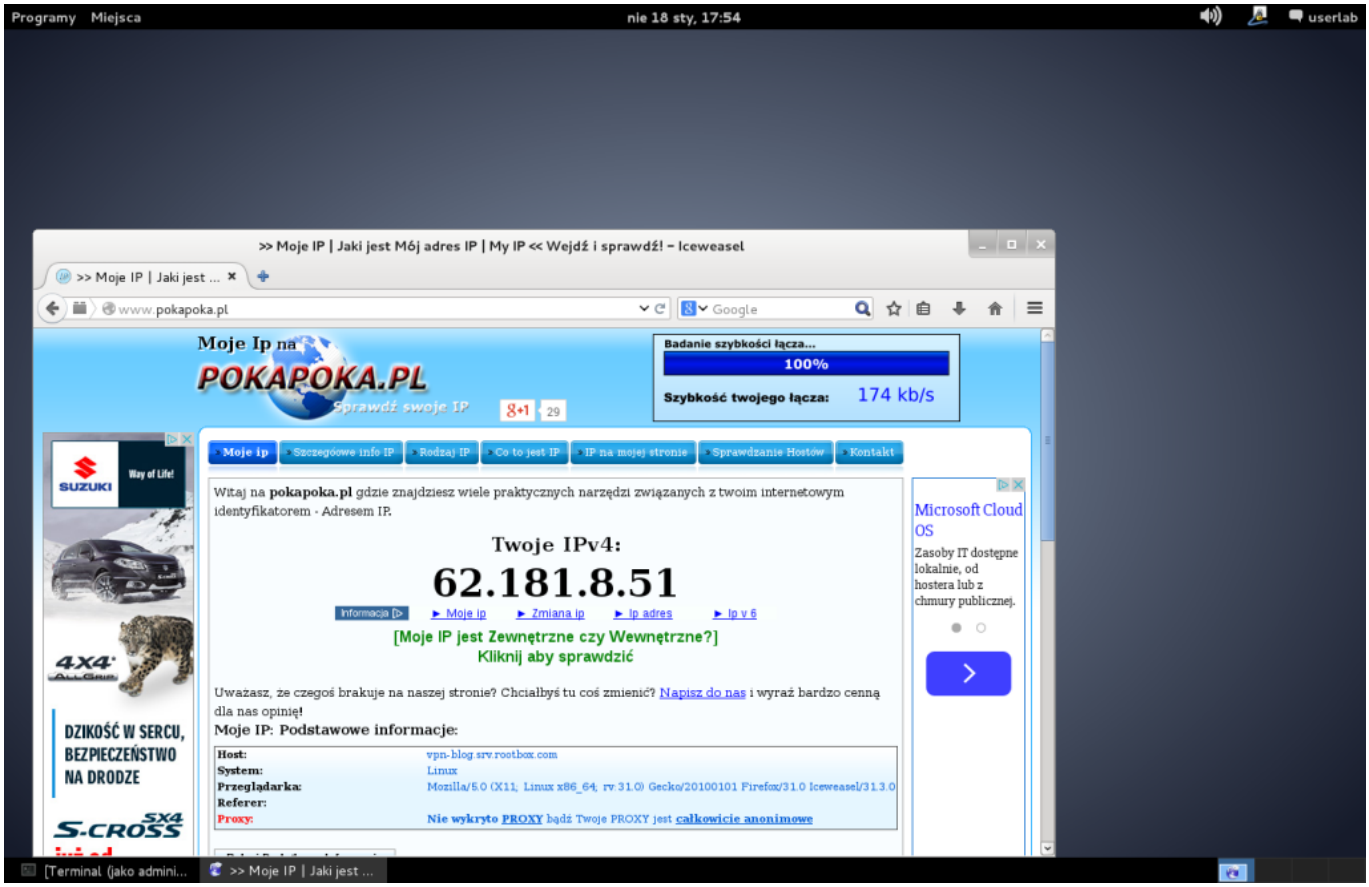

Uruchamiamy połączenie VPN i czekamy na efekty




Po chwili ukazuje się naszym oczom komunikat o tym, że połączenie zostało nawiązane.




Ponownie sprawdzamy ip i widzimy, że identyfikujemy się za pomocą ip VPN’a, wiec udało się nam.




## Pro-tip na zwiększenie bezpieczeństwa.

Domyślacie się, że tak naprawdę najsłabszym ogniwem naszego vpna jest port 22 wystawiony na public. Zmienimy więc troche ustawienia ssh.

```
echo "ListenAddress 6.6.6.1" >> /etc/ssh/sshd_config
netstat -tlpn
```

Teraz do ssh mamy dostęp jedynie podczas połączenia vpn i tylko wtedy.

## Atak MITM.

O MITM już pisałem w jednym z poprzednich artykułów. Dziś ten sam atak przeprowadzimy na ofiarę, która jest połączona do VPN'a. Jak ustaliliśmy w poprzednim artykule o MITM,

w sytuacji gdy intruz i ofiara są w tej samej sieci atak ten jest łatwy do wykonania.

Kilka danych na początek.

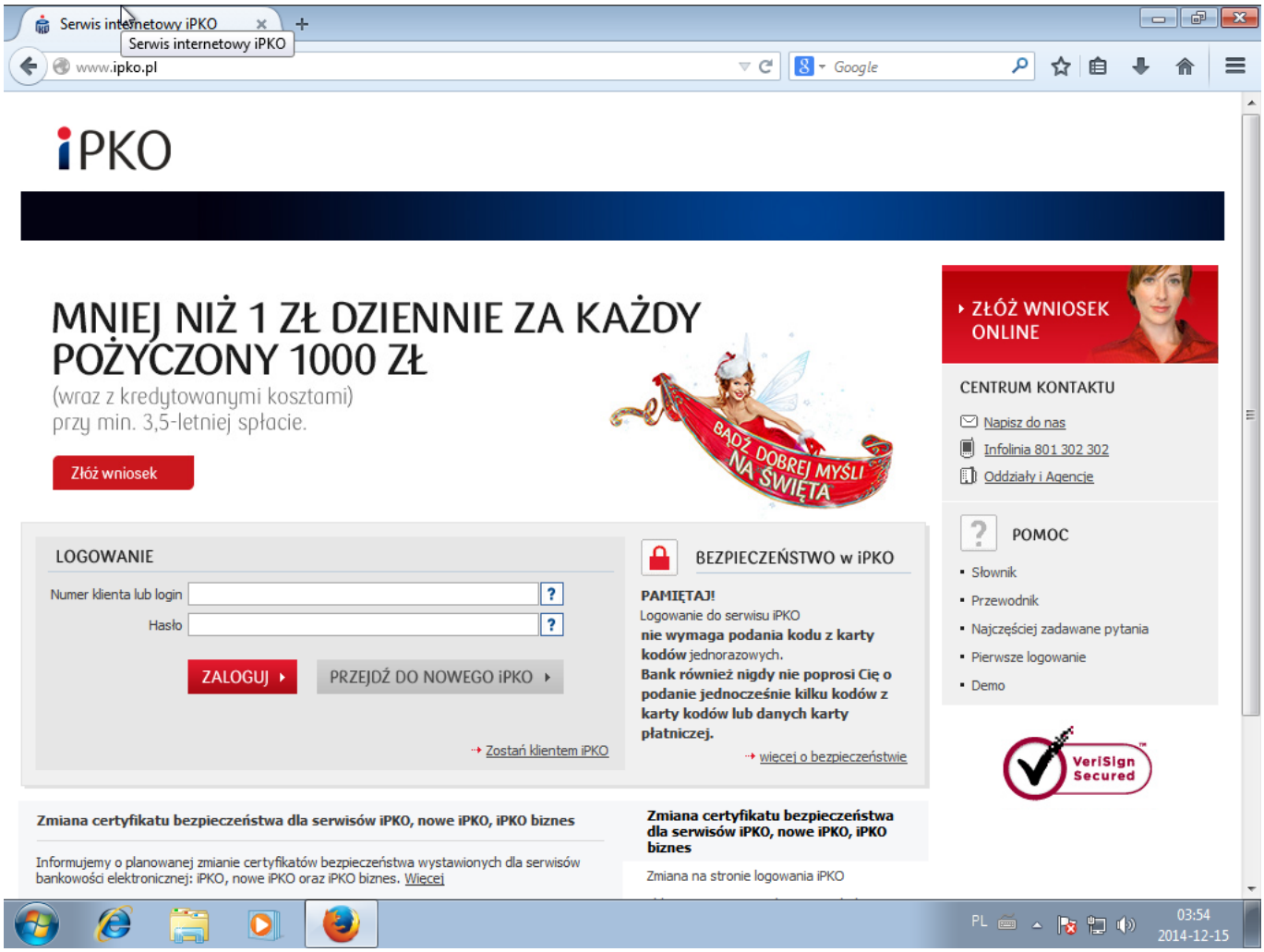
IP Routera: 10.0.2.1

IP Ofiary: 10.0.2.6

IP Intruza: 10.0.2.8

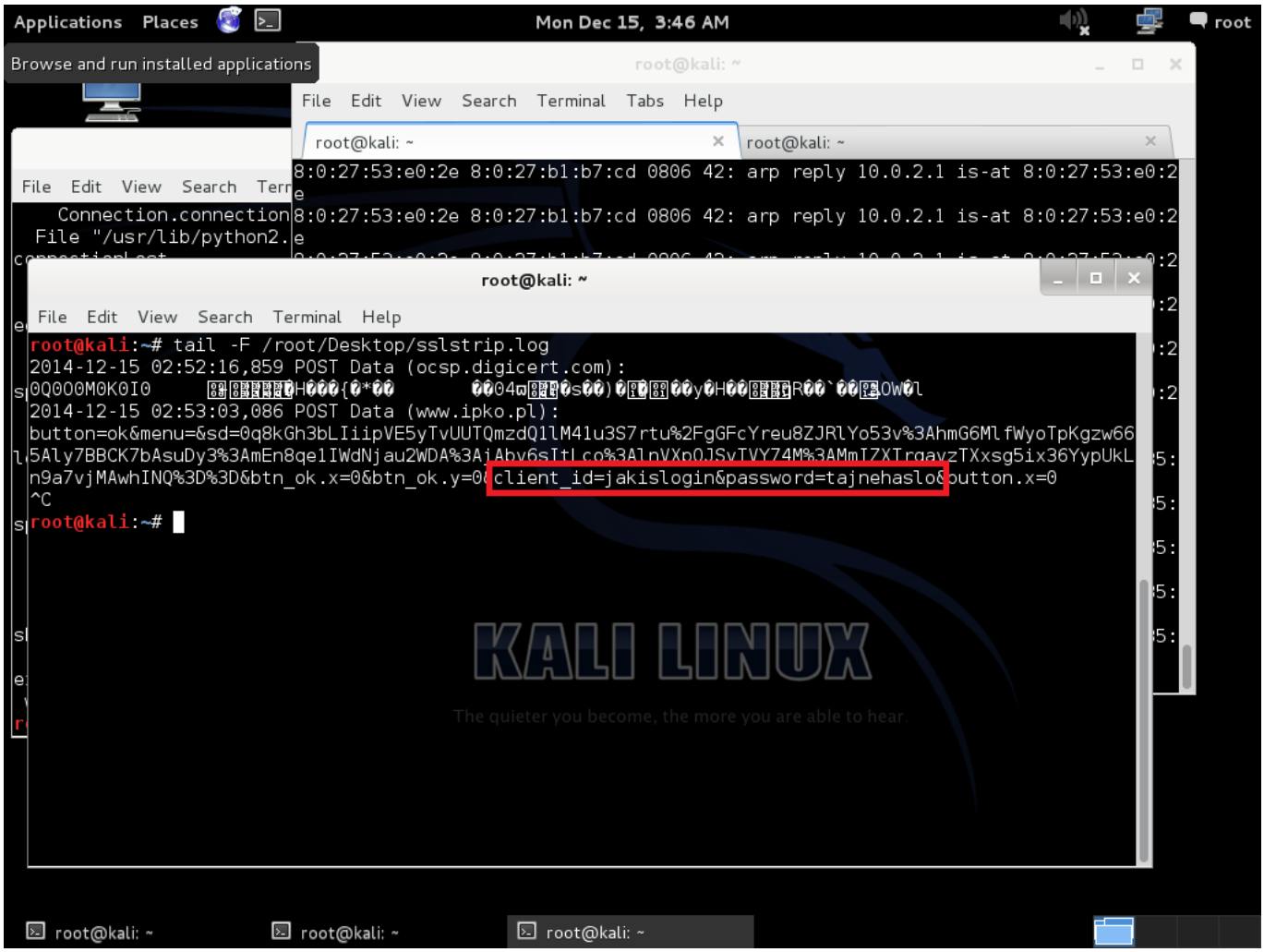
Pozwolę sobie pominąć opis samego ataku, przedstawię rezultaty. Dla porównania:

Gdy ofiara próbuje się zalogować na stronę banku gdy jest połączona bez vpn'a a intruz atakuje metodą MITM.

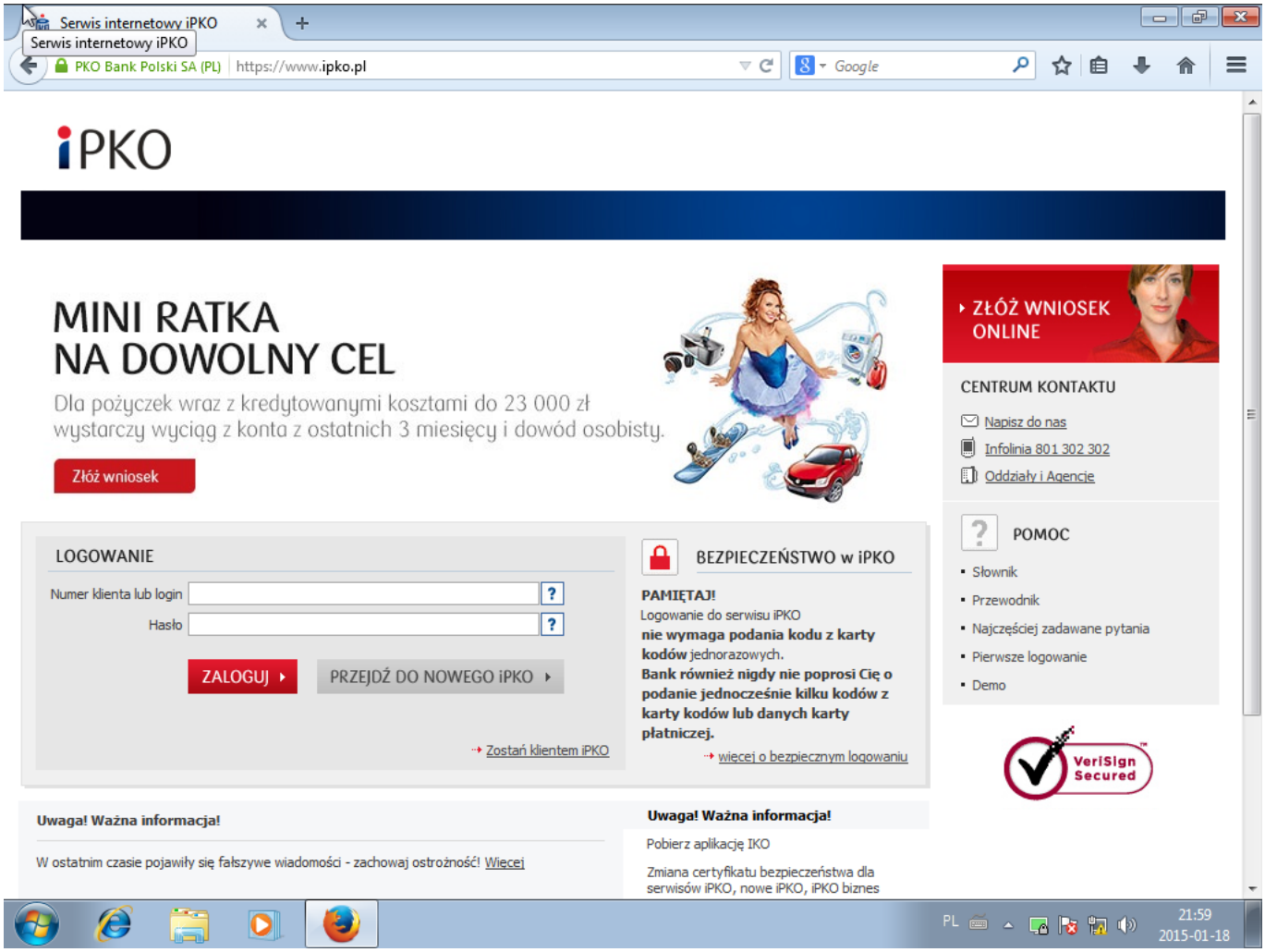


Wyraźnie widać, że połączenie nie jest zabezpieczone za pomocą protokołu SSL (brak zielonego paska.)

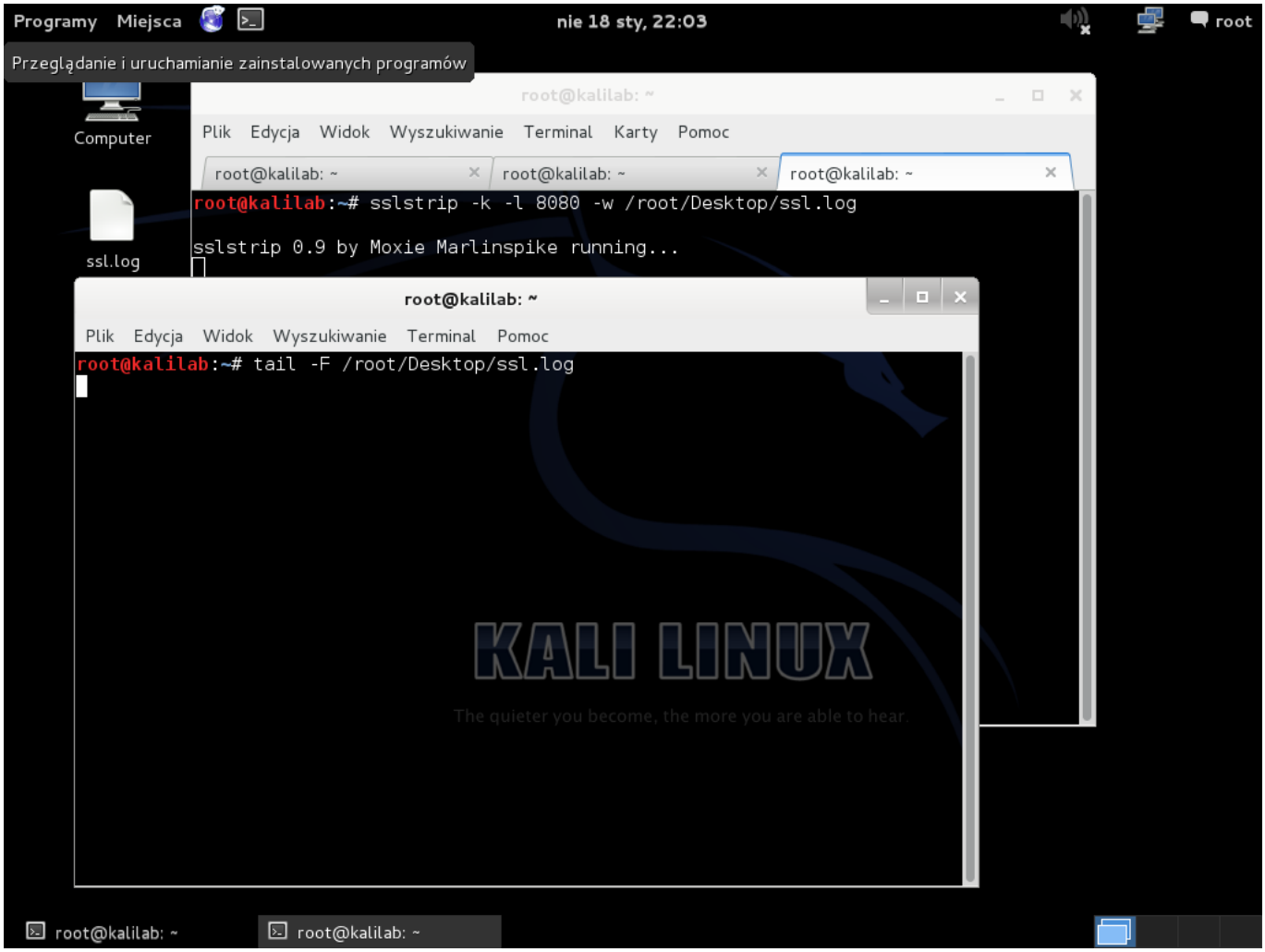
Oraz widok maszyny atakującego podczas próby zalogowania do banku.




Teraz ta sama sytuacja tylko, że ofiara używa VPN’a.




Już na pierwszy rzut oka widzimy zielony pasek świadczący o tym, że nasze połączenie jest kierowane prosto do banku.




Jak widzimy logi intruza są puste. Pomimo iż atakuje ofiarę, cały ruch odbywa się za pomocą tunelu VPN.

Oczywiście, VPN nie jest 100% sposobem na bezpieczeństwo. Bezpieczna jest tylko komunikacja pomiędzy klientem a serwerem.

Myślę, że ten artykuł pokazał Wam, jak skonfigurować własny VPN, oraz, że warto go mieć.