Projekt: Odporność Na Inwigilację (ONI v.1.0)

Spis treści

[2. Opis projektu 1](#_Toc66995465)

[3. Najważniejsze cechy 2](#_Toc66995466)

[4. Kluczowe elementy projektu 2](#_Toc66995467)

[5. Urządzenia 2](#_Toc66995468)

[6. Harmonogram projektu 2](#_Toc66995469)

[7. Architektura VPN 4](#_Toc66995470)

[8. Architektura DFS 5](#_Toc66995471)

[9. Ścieżka konfiguracji środowiska 5](#_Toc66995472)

[Wireguard 5](#_Toc66995473)

[Syncthing 6](#_Toc66995474)

[Rsnapshot 7](#_Toc66995475)

[10. ONI v.2.0 8](#_Toc66995476)

# Opis projektu

Niniejszy projekt jest projektem koncepcyjnym, którego pierwsza wersja v.1.0 jest wersją beta (nie produkcyjną). W kolejnym etapie projektu powstanie wersja v.2.0, która będzie zawierała dodatkowe elementy oraz fazę testów, przez co stanie się pierwszą wersją w pełni gotową do bezpiecznej implementacji dla użytkownika (produkcyjna).

Projekt ma na celu umożliwienie użytkownikowi założenia własnego bezpiecznego magazynu danych, do którego jest możliwy dostęp zdalny. Dostęp do danych jest realizowany poprzez łącze szyfrowane (**szyfrowanie w ruchu**), a dane są składowane w należących do użytkownika maszynach, które tworzą węzły **klastra**. Klaster zapewnia możliwość rozproszonego składowania tych samych danych na różnych urządzeniach, dzięki czemu magazyn użytkownika uzyskuje redundancję i odporność na awarię. Użytkownik ma możliwość decydowania o lokalizacji składowania poszczególnych katalogów magazynu. Na wypadek problemów wynikających z błędnej konfiguracji oprogramowania tworzącego klaster i skutkujących utratą danych, jeden z węzłów klastra jest wyposażony w funkcję backup’u w postaci snapshotów. Snapshot’y są składowane na oddzielnym dysku HDD i zawierają całość obrazu magazynu danych.

W ramach konfiguracji użytkownik otrzymuje również **bezpieczną sieć VPN**, do której są podłączone wszystkie urządzenia będące węzłami klastra. Wszelkie połączenia wychodzące do Internetu z urządzeń użytkownika są szyfrowane, a odszyfrowywanie odbywa się dopiero na serwerze VPN. Tam też są podmieniane adresy IP urządzeń użytkownika na adres serwera VPN.

# Najważniejsze cechy

1. Dostęp do Internetu poprzez bezpieczną sieć VPN z każdego urządzenia użytkownika.
2. Ochrona przed inwigilacją ze strony dostawcy Internetu i wszelkich punktów dostępu do sieci np. publiczne router’y, sieć komórkowa.
3. Brak możliwości namierzenia prawdziwego IP użytkownika przez dostawcę usługi w Internecie dzięki podmianie adresu IP na serwerze VPN.
4. Bezpieczne składowanie danych o wysokim priorytecie bez ryzyka ich utraty.
5. Wersjonowanie plików.
6. Realna możliwość całkowitej rezygnacji z usługi kopii zapasowej świadczonej przez gigantów technologicznych (Google Drive, Microsoft Onedrive, Dropbox itp.) i tym samym uniemożliwienie takim podmiotom czytania i analizowania osobistych danych.
7. Synchronizacja danych pomiędzy urządzeniami z wykorzystaniem protokołu p2p, który umożliwia upload danych z kilku węzłów jednocześnie, co niweluje typowy problem wąskiego gardła dla magazynów danych związany z przepustowością upload’u z pojedynczego węzła.

# Kluczowe elementy projektu

1. Sieć prywatna (Virtual Private Network: **VPN**)
2. Rozproszony magazyn plików w chmurze prywatnej (Distributed File Storage: **DFS**)
3. Anonimizacja endpointu sieci prywatnej VPN (Anonymized Virtual Private Network: **AVPN**)

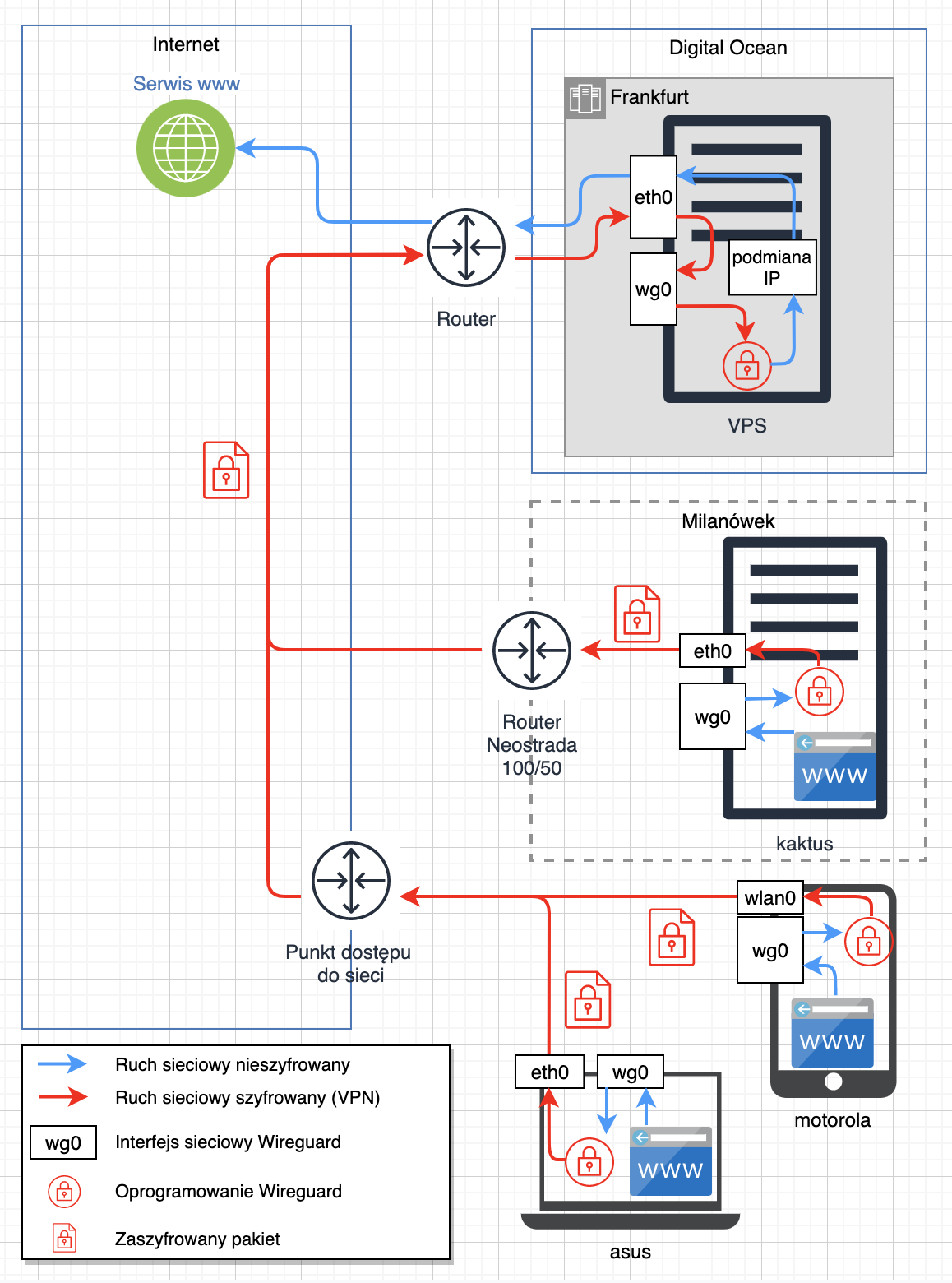
# Urządzenia

|  |  |
| --- | --- |
| **Nazwa urządzenia** | **Informacje** |
| kaktus | Ubuntu 20.04.2, 256 GB SSD, 460 GB HDD |
| motorola | Android 10, 128 GB UFS 2.1, 256 GB MicroSD V30/A1 |
| asus | Windows 10, 256 GB SSD |

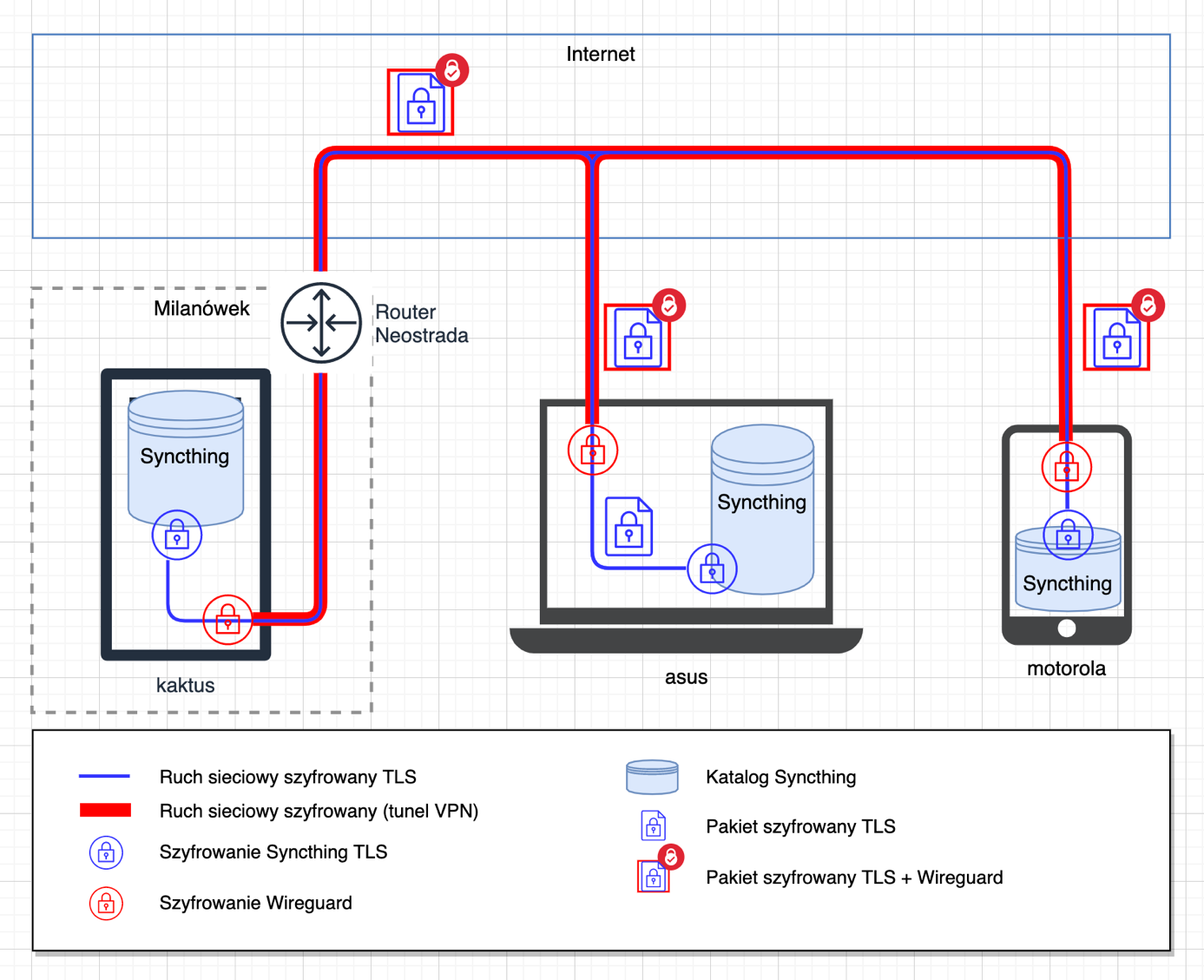
# Harmonogram projektu

|  |  |  |
| --- | --- | --- |
| **Element projektu** | **Zadanie** | **Status** |
| VPN | Instalacja i konfiguracja oprogramowania Wireguard na serwerze VPS | Zakończono |
| VPN | Instalacja i konfiguracja oprogramowania Wireguard na endpointach klientów:   1. motorola 2. asus 3. kaktus | Zakończono |
| VPN | Test: podmiana IP na dla wszystkich klientów sieci VPN | Zakończono |
| VPN | Konfiguracja automatycznego uruchomienia klienta Wireguard przy starcie systemu na magazynie głównym | Zakończono |
| DFS | Instalacja i konfiguracja oprogramowania Syncthingna magazynie głównym klastra DFS:   1. kaktus | Zakończono |
| DFS | Instalacja i konfiguracja oprogramowania Syncthing u klientów klastra DFS:   1. motorola 2. asus | Zakończono |
| DFS | Nadanie węzłom klastra DFS Syncthing adresów sieci prywatnej VPN | Zakończono |
| DFS | Test: połączenie sieciowe i synchronizacja katalogów między urządzeniami | Zakończono |
| DFS | Test: pobieranie danych z dwóch urządzeń jednocześnie | Zakończono |
| DFS | Konfiguracja automatycznego uruchomienia klienta Syncthing przy starcie systemu na magazynie głównym | Zakończono |
| DFS | Wyłączenie funkcji *relay* na wszystkich węzłach | Zakończono |
| DFS | opracowanie strategii przywracania Syncthing po awarii na magazynie głównym: Reboot | Zakończono |
| DFS | Test: Maksymalny rozmiar plików: 53 GB | Zakończono |
| DFS | Ustawienie maksymalnych wartości dla ruchu wychodzącego i przychodzącego na urządzeniach: 10MBs / 10MB/s | Zakończono |
| DFS | Rozważenie szyfrowania https/tls dla GUI: Nie dla wersji v.2.0 | Zakończono |
| DFS | Zaawansowane/Options/Always Local Nets | Zakończone |
| DFS | Zaawansowane/Devices/Addresses: dynamic | Zakończone |
| DFS | Zaawansowane/Allowed Networks: 10.10.10.0/24 | Zakończone |
| DFS | opracowanie strategii Auto Accept Folders dla magazynu głównego | Zakończone |
| DFS | kopia zapasowa: Montaż dodatkowego dysku HDD w kaktusie do celów kopii zapasowej dla magazynu głównego | Zakończone |
| DFS | kopia zapasowa: Instalacja i konfiguracja kopii zapasowych całości obrazu magazynu głównego na dodatkowy dysk HDD | Zakończone |
| DFS | Niezależność software’owa pomiędzy dyskiem kopii zapasowej a magazynem głównym: Syncthing i Rsnapshot | Zakończone |
| DFS | kopia zapasowa: ustalenie harmonogramu tworzenia kopii zapasowej magazynu głównego | Zakończone |
| DFS | kopia zapasowa: automatyzacja procesu | Zakończone |
| AVPN | Powołanie serwera VPS korzystając z anonimowego adresu email oraz płatności z wykorzystaniem kryptowaluty | Wycofane z wersji v.1.0 |
| AVPN | Instalacja i konfiguracja oprogramowania Wireguard na anonimowym serwerze VPS | Wycofane z wersji v.1.0 |
| AVPN | Instalacja i konfiguracja oprogramowania Wireguard na endpointach klientów:   1. motorola 2. asus 3. kaktus | Wycofane z wersji v.1.0 |
| AVPN | Test: podmiana IP dla wszystkich klientów VPN | Wycofane z wersji v.1.0 |

# Architektura VPN



# Architektura DFS



# Ścieżka konfiguracji środowiska

## Wireguard

1. Instalacja Wireguard na serwerze i kliencie:

*sudo apt install wireguard*

1. Wygenerować klucze:

*wg genkey | tee privatekey | wg pubkey > publickey*

1. Tworzenie pliku konfiguracyjnego na serwerze VPN. Wejść do katalogu /etc/wireguard, i utworzyć tam plik wg0.conf, edytować go do takiej postaci:

[Interface]

Address = 10.10.10.1/32

SaveConfig = true

PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE; ip6tables -A FORWARD -i wg0 -j ACCEPT; ip6tables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE; ip6tables -D FORWARD -i wg0 -j ACCEPT; ip6tables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

ListenPort = 51820

PrivateKey = xxx

[Peer]

PublicKey = xxx

AllowedIPs = 10.10.10.2/32

1. Tworzenie pliku konfiguracyjnego na kliencie VPN. Wejść do katalogu /etc/wireguard, utworzyć plik wg0.conf, edytować go do takiej postaci:

[Interface]

Address = 10.10.10.20/24

PostUp = echo nameserver 1.1.1.1 | resolvconf -a tun.%i -m 0 -x

PostDown = resolvconf -d tun.%i

PrivateKey = GL52vrLk50Wt3mGVBg8RNeJghY/qLo0Kf14i+e/qFXU=

[Peer]

PublicKey = hrZFDhQo9meMDrX03ivzDXPxT9qNw67vZer/E9jTIl8=

Endpoint = 206.81.30.177:51820

AllowedIPs = 0.0.0.0/0

1. Po odpowiednim wpisaniu private & public kluczy oraz adresów IP wykonać komendy na serwerze i kliencie:
   1. *wg-quick up wg0*
   2. *host myip.opendns.com resolver1.opendns.com*
2. Ustawić automatyczne uruchamianie przy reboocie:

*systemctl enable wg-quick@wg0*

## Syncthing

1. Pliki konfiguracyjne, ścieżki:

*syncthing -paths*

1. Zainstalować Syncthing:

*sudo apt install Syncthing*

1. Uruchomić Syncthing:

*syncthing -gui-address="10.10.10.20:8384"*

1. Z urządzenia w sieci wewnętrznej VPN wejść z przeglądarki na adres 10.10.10.20:8384 (lub inny odpowiedni podany jako atrybut -gui-adress) i skonfigurować Syncthing:
   1. Zmienić adres serwera na 10.10.10.20:8384
   2. wyłączyć relay
   3. Always local nets: 10.10.10.0/24
   4. Device/Allowed Networks: 10.10.10.0/24
2. Można ewentualnie przejść do katalogu /.config/syncthing/config.xml i edytować plik konfiguracyjny.
3. Ustawić automatyczne uruchamianie przy reboocie:
   1. wyłączyć syncthing
   2. zainstalować supervisor:

*sudo apt install supervisor*

* 1. wejść do katalogu: /etc/supervisord/conf.d/ i utworzyć plik o nazwie syncthing.conf, a następnie wkleić do niego poniższą zawartość wstawiając odpowiedniego użytkownika zamiast *<USERNAME>:*

*[program:syncthing]*

*autorestart = True*

*directory = /home/<USERNAME>/*

*user = <USERNAME>*

*command = /usr/bin/syncthing -no-browser -home="/home/<USERNAME>/.config/syncthing"*

*environment = STNORESTART="1", HOME="/home/<USERNAME>"*

* 1. wykonać komendę:

*supervisorctl reload*

* 1. sprawdzić czy działa:

*supervisorctl status syncthing*

* 1. sprawdzić czy działa po reboocie:

*sudo reboot*

## Rsnapshot

1. Zainstalować rsnapshot:

*sudo apt install rsnapshot*

1. Przejść do katalogu /etc i edytować plik rsnapshot:

*sudo nano /etc/rsnapshot.conf*

* 1. root directory: snapshot\_root /katalog/w/którym/będzie/magazynowany/backup
  2. BACKUP LEVELS / INTERVALS (harmonogram snapshot’ów, w poniższej konfiguracji magazynowane będą snapshoty codzienne do 6 dni wstecz, cotygodniowe do 5 tygodni wstecz i comiesięczne do 6 miesięcy wstecz):
     1. retain daily 7
     2. retain weekly 5
     3. retain monthly 6
  3. LOCALHOST (czyli co będzie backup’owane):
     1. backup /katalog/podlegający/backup’owi localhost/subdirectory/(patrz 2.a.)

1. Przetestować, czy składnia konfiguracji jest poprawna:

*sudo rsnapshot configtest*

1. Przetestować, czy backup jest prawidłowo skonfigurowany:

*sudo rsnapshot daily*

1. Zaplanować snapshoty poprzez cron.d:
   1. edytować plik /etc/cron.d/rsnapshot (w poniższej konfiguracji snapshot będzie robiony codziennie o 6 rano, 1-go dnia tygodnia o 6 rano i 1-go dnia miesiąca o 6 rano):
      1. 0 6 \* \* \* root /usr/bin/rsnapshot daily
      2. 0 6 \* \* 1 root /usr/bin/rsnapshot weekly
      3. 0 6 1 \* \* root /usr/bin/rsnapshot monthly

# ONI v.2.0

Wersja v.2.0 przewiduje uzyskanie stabilności i bezpieczeństwa działania systemów AVPN i DFS w stopniu umożliwiającym taką implementację przez użytkownika, która pozwoli na stałe zrezygnowanie z usług magazynowania danych oferowanych przez dostawców chmury publicznej. Planowane w wersji v.2.0 jest w szczególności:

1. Ukończenie anonimowej sieci AVPN
2. Ukończenie 30-dniowego testu działania systemów VPN i DFS w celu dowiedzenia ich stabilności
3. Ustalenie procedury tworzenia archiwum w interwale ok. 6 mies. na zewnętrznym nośniku
4. Ustalenie procedury tworzenia archiwum szyfrowanego w zewnętrznej chmurze na wypadek utraty dostępu do wszystkich własnych urządzeń