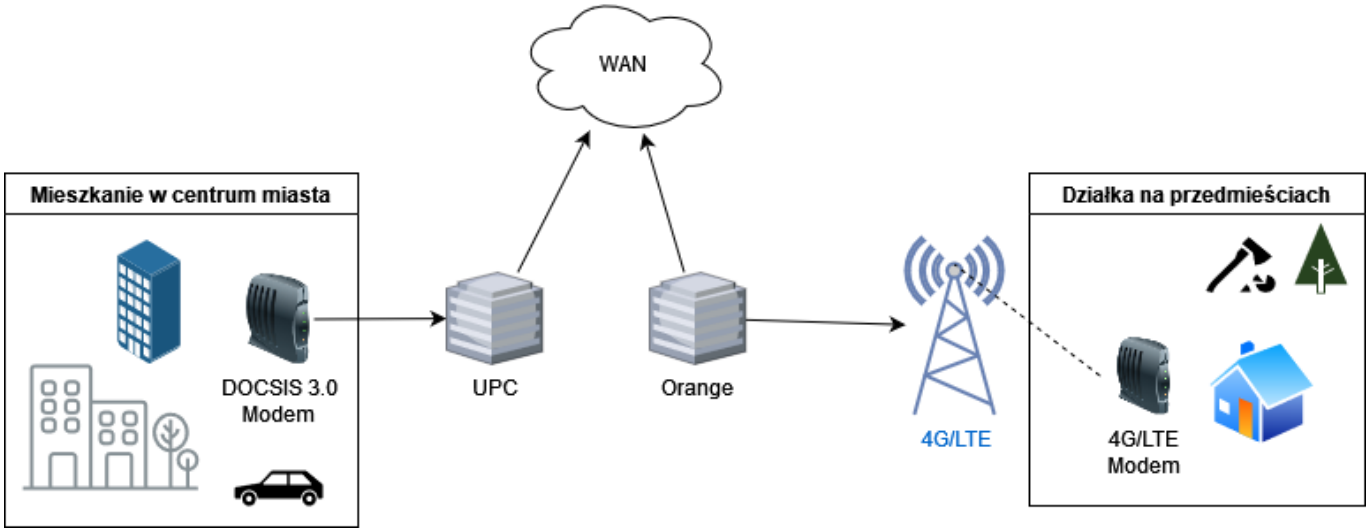
Network: Projekt Home-Lab'a 1.0

Wymagania i specyfikacja

Moja sieć domowa to cały czas zmieniająca się struktura zależna od tego co akurat robię dla danego klienta lub co akurat mnie zainteresowało i postanowiłem to przetestować na żywym organizmie. Co do moich wymagań to nie są wielkie, użytkowników sieć będzie miała no najwyżej kilka, z czego jeden to ja 😉. Sieć musi być bardzo możliwie konfigurowalna niektóre usługi udostępnione przez tunele natomiast inne przez przekierowanie portów i reverse proxy. Niektóre usługi muszą być dostępne z zewnątrz, dla mnie ale również dla klientów i dla osób dla których prowadzę szkolenia. Innym aspektem który jest dla mnie ważny to optymalizacja kosztów, urządzenia muszą być w miarę możliwości tanie tak żeby można było je łatwo wymienić. Sporo z komponentów tej sieci którą wam tutaj przedstawie to sprzęt klasy domowej lub został przeze mnie odkupiony (za grosze) od niegdyś moich klientów.

Schemat lokalizacji sieci



Wymagania

Wymagania co do mojej sieci:

- publiczny adres IPv4 (może być dynamiczny)
- brak blokad na tunele i IPsec od ISP
- łatwa i szybka konfigurowalność
- możliwość zestawienia tunelu VPN pomiędzy dwoma lokalizacjami (tylko do monitoringu)
- hostowanie usług na różnych domenach
- urządzenia i systemy z dobrą dokumentacją
- małe koszty

Specyfikacja

ISP

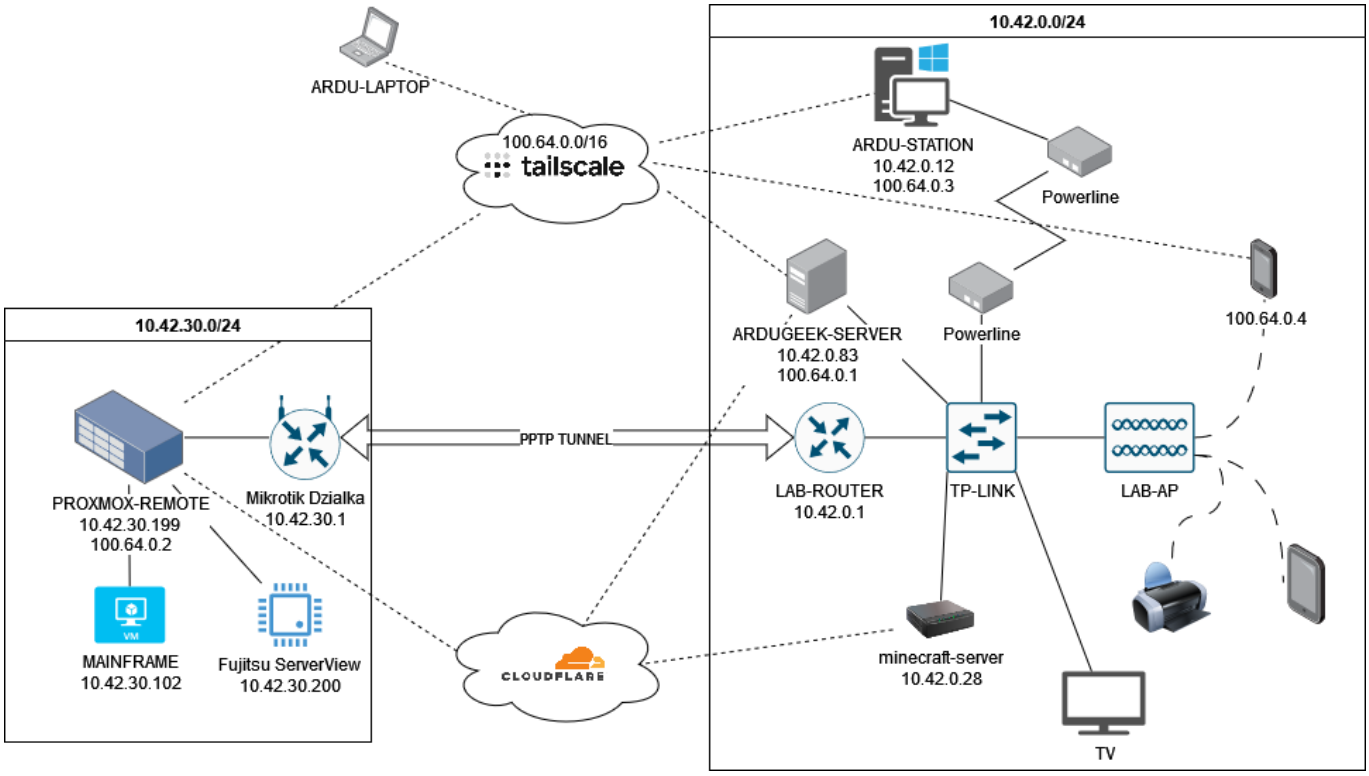
Jeżeli chodzi o mojego dostawcę internetu jest to PLAY (niegdyś UPC). Każdy abonent sieci UPC dostaje dynamiczny adres IPv4, z czasem dzierżawy około miesiąca, więc wyczerpuje to moje wymagania co do adresu IP. Modem dostarczony przez UPC to pudełeczko o nazwie „Horizon”, jest to zwykły modem w standardzie DOCSIS 3.0 (Bardzo fajny materiał który pokazuje jak działa DOCSIS 1.0/1.1).

Routery, Switche i inne urządzenia sieciowe

Zdjęcie:				
Opis	Mój router: MikroTik hAP AX lite	Mój switch: TPlink TL-SG105E	Mój accesspoint: Mikrotik mAP lite	Router w drugiej lokalizacji: Mikrotik RB941
Nazwa	LAB-ROUTER	TP-LINK	LAB-AP	MikrotikDzialka

Tak jak wspomniałem we wstępie nie są to urządzenia klasy Enterprise, natomiast na moje potrzeby wystarczają i pozwalają na wysoką konfigurowalność.

Schemat sieci



Opis schematu

Jak widać powyżej sieć nie jest bardzo skomplikowana ale ma klika głównych komponentów takich jak serwery i routery w obywdu lokalizacjach. W następnych sekcjach omówimy sobie poszczególne elementy całego schematu.

LAB-ROUTER

To jest główny router na którym są wszystkie przekierowania portów na odpowiednie serwery. Kiedyś był on również serwerem VPN, teraz już tylko dostał tunel PPTP do monitorowania urządzeń w drugiej lokalizacji. Przekierowania portów są do usług na serwerach minecraft-server i ardugeek-server. Wykorzystuję na nim funkcję DDNS dostępną od mikrotika gdyż nie wszystkie usługi chce mieć na tunelach cloudflare.

Mikrotik Dzialka

To jest tylko prosty klient dla sieci 4G/LTE jako klient DHCP z bramą. Skonfigurowany jest jako klient tunelu PPTP. Tunel ten na dzień dzisiejszy wykosztuję tylko do monitorowania urządzeń. Całą resztę robi tailscale. Poniżej jest sprzęt jaki wykorzystuję do łączności 4G/LTE

TP-LINK TL-MR100
Źródło: Morele.Net

Antena 4G/LTE
Źródło: Morele.Net

PROXMOX-REMOTE - Serwer fizyczny

Fizycznie jest to stary serwer Fujitsu Primergy RX300 S8, który dawno temu odkupiłem od jednego z moich klientów. Mam na nim zainstalowanego hyper-visora Proxmox VE. Nie jest to serwer z wygórowaną specyfikacją ale wystarcza na różne eksperymenty. Ma procesor Intel Xeon, 32GB pamięci ram, Dwie macierze jedną RAID1 (256GB SSD) i drugą RAID5 (3.5TB HDD). Na nim jest

uruchomiona usługa tailscale.

Jest to też węzeł wyjściowy dla sieci 10.42.30.0/24, więc jeżeli który kolwiek z klientów sieci Tailscale będzie chciał się połączyć z czymkolwiek w tej sieci to wyjdzie do niej przez ten serwer.

MAINFRAME - Maszyna wirtualna na proxmox-remote

Logo dystrybucji Xubuntu
Źródło: wikimedia.org

Jest to maszyna wirtualna z systemem xubuntu 22.04. Na niej mam uruchomione usługi takie jak:

- https://ai.ardugeek.ovh/ - Open Web UI
- https://jupyter.ardugeek.ovh/ - Jupyter lab
- https://kasm.ardugeek.ovh/ - Kasm workspaces
- https://portainer.ardugeek.ovh/ - Portainer
- https://files.ardugeek.ovh/ - File browser
- https://shell.ardugeek.ovh/ - Shell in a box

Wszystkie z tych usług są hostowane za pomocą tunelu cloudflare. Nie mam dostępnego adresu publicznego w tej lokalizacji na karcie sim, więc niestety pozostaje tylko tunel cloudflare. Nie są to usługi dla mnie krytyczne więc jak tunel się rozłączy (a ma tendencję raz na jakichś czas 😉) to nie ma to wpływu na działanie mojej sieci. Connector tailscale nie działa na tej maszynie wirtualnej tylko działa na samym hypervisorze.

ARDUGEEK-SERVER - Serwer fizyczny

Jest to fizyczny komputer który pewien czas temu złożyłem z zamówionych komponentów, ma on jak na dzisiejsze czasy już trochę przestarzałą specyfikację ale na moje potrzeby wystarcza w 100%.

System na nim zainstalowany to Windows Server 2019. Specyfikacja tego serwera to: Intel i3 3 generacji, 16GB pamięci ram, i wiele różnego rodzaju dysków zewnętrznych i wewnętrznych podłączonych do niego. Wiele z nich jest backupowanych w różne miejsca (inne dyski lub onedrive) za pomocą narzędzia Duplicati z retencją i różnicowe. Jest to centralny punkt całej mojej sieci, jest to serwer od którego wszystko się zaczęło (oczywiście nie ten konkretny, było wiele jego poprzedników o tej samej nazwie), to do niego się łączę zdalnie jak jestem gdzieś poza domem a coś się zepsuło i muszę coś naprawić zdalnie, to na nim testuję różne rozwiązania, to na nim jest uruchomiona ta Wiki. Wszystko co jest w mojej sieci jest w jakichś sposób powiązane z tym serwerem.

Jest to też węzeł wyjściowy dla siecio 10.42.0.0/24 lub dla całego ruchu w sieci tailscale. Pozwala to klientom sieci Tailscale wychodzić na świat przez ten komputer.

Poniżej zestawienie usług działających na tym serwerze.

Usługi wewnętrzne nie dostępne publicznie:

- Node-Red
- Microsoft SQL server
- Adminer
- Duplicati
- WakeMeOnLan

Usługi dostępne publicznie:

- https://start.ardugeek.ovh/ - Strona startowa do moich usług, statyczny HTML/CSS
- https://ostrowski.net.pl/ - Moje strona wizytówka z informacją o szkoleniach i kontaktem do mnie, statyczny HTML/CSS
- https://wiki.ostrowski.net.pl/ - Wiki którą właśnie czytasz skonfigurowana za pomocą pakietu DokuWiki
- https://filevista.ardugeek.ovh/ - Serwer plików wykorzystywany do moich prywatnych celów ale i dla klientów, skonfigurwany za pomocą narzędzia FileVista
- https://plex.ardugeek.ovh/ - Serwer plex do multimediów. (jestem chyba jednym z ostatnich klientów którzy kupują multimedia w empiku na płytach albo w formacie MP3/MP4)
- https://audiobook.ardugeek.ovh - Audiobookshelf aplikacja do słuchania audiobooków
- https://auth.ardugeek.ovh - Skrypt mojego autorstwa pozwalający na bezpieczne łączenie się przez translację NAT do usług.

ARDU-STATION - komputer fizyczny

To jest moja główna stacja robocza i komputer na którym zdarza mi się ogrywać gry wideo. Specyfikacja to Ryzen 5 5800U, 32GB ram i 1TB SSD, 1TB HDD, 2TB HDD. To jest główne stanowisko z całym środowiskiem do wirtualizowania i testowania nowych rozwiązań to tutaj wykosztuję środowisko Hyper-V i CHR od MikroTika do szkoleń. To tutaj powstał i jest wykorzystywany skrypt do mapowania topologii Hyper-V. Tutaj również mam zainstalowany konnector tailscale ponieważ stąd również się łączę do moich serwerów i testuję różne rozwiązania a w wolnym czasie lubię pograć w klasyki gatunku.

Tutaj usług jako takich nie ma mogę w sumie wymienić tylko jedną mianowicie mam tutaj WSL2 skonfigurowany tak żebym mógł na nim uruchomić program GNU-RADIO do symulacji przetwarzania sygnałów.

minecraft-server - serwer fizyczny

To jest bardzo stary przemysłowy mini pc od firmy giada. Ma tylko 8GB pamięci ram i intela i3 3gen. Tutaj mam zainstalowany pakiet Application Managment Panel od CubeCoders, hostuję tutaj serwery do grania ze znajomymi i rodziną. Głównie na ten moment działa tutaj wieloplatformowy serwer Minecraft. Mam tutaj uruchomione zadanie w Cron'ie które co dwa dni pakuje cały serwer Minecraft w ZIP i wysyła na ardugeek-server. Konfigurację mojego serwer możesz zobaczyć tutaj.

Jedyne co jest dodatkowe tutaj poza standardową konfiguracją serwer Minecraft jest to że tunel cloudflare udostępnia poniższe witryny:

- https://amp.ardugeek.ovh - panel do zarządzania serwer Minecraft
- https://bluemap.ardugeek.ovh - plugin BlueMap do wyświetlania mapy serwera w przeglądarce

Plugin Bluemap

> BlueMap is a program that reads your Minecraft world files and generates not only a map, but also 3D-models of the whole surface. With the web-app you then can look at those in your browser and basically view the world as if you were ingame! Or just look at it from far away to get an overview.

Z https://bluemap.bluecolored.de/

Jest to moje niedawne odkrycie i uważam że ten plugin powinien mieć każdy kto ma swój własny serwer Minecraft. Plugin ten pozwala w formie 3D przeglądać świat gry tak jakbyśmy byli w grze. Plugin na szczęscie jest napisany w JavaScript'cie więc nie obciąża serwera. Wygląda to spektakularnie. Instalacja jest prosta, a port na jakim mapa jest widoczna to `8100`.

Zasada działania Tailscale

Tailscale to usługa VPN oparta na architekturze mesh, która upraszcza łączenie wielu urządzeń w bezpiecznej sieci prywatnej (tailnet). Zamiast tradycyjnego modelu hub-and-spoke (gdzie cały ruch przepływa przez centralny serwer), Tailscale umożliwia bezpośrednie, szyfrowane połączenia punkt-punkt między węzłami sieci. Każde połączenie jest szyfrowane end-to-end za pomocą protokołu WireGuard (z algorytmem ChaCha20-Poly1305), a klucze prywatne pozostają wyłącznie na urządzeniach użytkowników. Węzły sieci wymieniają się kluczami i metadanymi przez płaszczyznę kontrolną – serwery koordynacyjne Tailscale – wykorzystując protokół Noise IK (X25519) opakowany w TLS, co umożliwia uwierzytelnianie i dystrybucję konfiguracji (ACL).

Dzięki mechanizmom NAT traversal każdy klient Tailscale próbuje przebić się przez zaporę (hole-punching) przy użyciu niestandardowego protokołu UDP. Jeśli połączenie bezpośrednie nie może zostać nawiązane (np. z powodu symetrycznego NAT), Tailscale automatycznie korzysta z relayów DERP (Designated Encrypted Relay for Packets) – dedykowanych serwerów przesyłających zaszyfrowane pakiety między węzłami. Dzięki temu możliwa jest łączność nawet w trudnych

warunkach sieciowych.

Kluczowe technologie i protokoły

- WireGuard – bazowy protokół VPN UDP używany do szyfrowania pakietów między urządzeniami.
- DERP – system serwerów relay pośredniczących w przekazie pakietów, gdy połączenie bezpośrednie zawiedzie.
- NAT traversal / hole punching – niestandardowy protokół UDP do przebijania się przez translatory adresów.
- Noise IK / TLS – protokoły szyfrowania i uwierzytelniania wykorzystywane w komunikacji z serwerami Tailscale.
- DNS i certyfikaty – MagicDNS i automatyczne certyfikaty TLS dla zasobów w tailnecie.

Zalety rozwiązania

- Bezpieczeństwo end-to-end – pełne szyfrowanie i prywatność, klucze prywatne nigdy nie opuszczają urządzeń.
- Brak konfiguracji sieciowej – działanie typu „zero-config”, łatwe dołączanie urządzeń.
- Niskie opóźnienie i wysoka wydajność – bezpośrednie połączenia peer-to-peer.
- Elastyczność – działa na wielu systemach operacyjnych, integruje się z Active Directory i chmurami.
- Prosta administracja – przez panel webowy można ustalać reguły ACL, definiować trasy, monitorować ruch.

Wady i ograniczenia

- Zaufanie do dostawcy – kontrola i koordynacja ruchu wymaga połączenia z serwerami Tailscale.
- Wymagania systemowe – konieczność instalacji klienta na każdym urządzeniu.
- Limity darmowego planu – większe sieci wymagają płatnego abonamentu.
- Wydajność w sieciach bez bezpośrednich połączeń – relay DERP może wprowadzać opóźnienia.
- Brak funkcji VPN do anonimizacji – Tailscale nie jest typowym VPN-em do omijania blokad czy maskowania IP.

Zastosowania praktyczne

- Zdalny dostęp do serwerów i urządzeń – np. SSH, RDP, kamery IP bez otwierania portów.
- Łączenie wielu lokalizacji – połączenia między biurami, chmurami i środowiskami dev/produkcyjnymi.
- IoT i homelab – bezpieczne połączenia z urządzeniami domowymi, Raspberry Pi itd.
- Zero Trust Networking – szczegółowa kontrola dostępu i segmentacja sieci.

Zasada działania Cloudflare Tunnel

Cloudflare Tunnel (wcześniej Argo Tunnel) umożliwia bezpieczne udostępnianie usług serwerowych (np. WWW, SSH, RDP) bez potrzeby posiadania publicznego IP ani otwierania portów. Na serwerze uruchamiany jest agent cloudflared, który inicjuje wyłącznie wychodzące, zaszyfrowane połączenia TLS do infrastruktury Cloudflare. Ruch użytkowników z Internetu trafia najpierw do Cloudflare, który przekazuje go przez tunel do serwera lokalnego.

Tunel można związać z nazwą domeny (np. CNAME), a Cloudflare zapewnia szyfrowanie, filtrowanie oraz ochronę przed atakami. Połączenia są utrzymywane przez HTTP/2 lub HTTP/3 (QUIC), co

zapewnia wysoką wydajność i odporność na opóźnienia.

Kluczowe technologie i protokoły

- cloudflared – agent na serwerze utrzymujący połączenie z Cloudflare.
- TLS 1.3 – zabezpiecza tunelowaną transmisję danych.
- HTTP/2 i HTTP/3 (QUIC) – protokoły transportowe wspierające wielowątkowość i niskie opóźnienia.
- Port 7844 / adresy Cloudflare – domyślny port komunikacji z Cloudflare; wystarczy tylko dostęp wychodzący.
- DNS i routing – tunel wiązany jest z nazwą DNS, co upraszcza udostępnianie zasobów.

Zalety rozwiązania

- Brak otwartych portów – wszystkie połączenia są wychodzące.
- Ochrona przed atakami – Cloudflare oferuje wbudowaną ochronę DDoS i firewall.
- Globalna sieć – dane przesyłane są najbliższą możliwą trasą do centrum danych Cloudflare.
- Prosta konfiguracja – szybka instalacja i integracja z panelem Cloudflare Zero Trust.
- Szerokie zastosowanie – obsługuje wiele typów usług (np. serwery WWW, SSH, RDP).

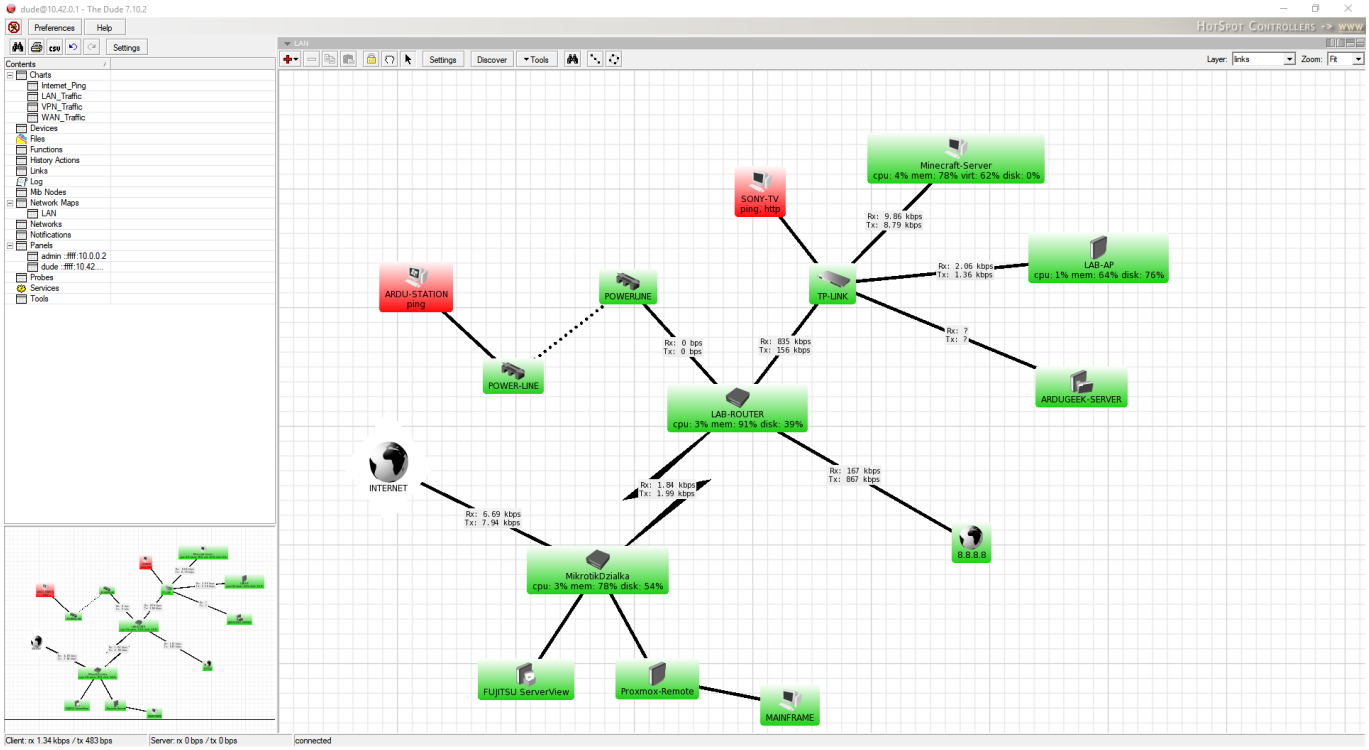
Wady i ograniczenia

- Zależność od Cloudflare – tunel działa tylko przez ich infrastrukturę.
- Prywatność – Cloudflare ma techniczny dostęp do odszyfrowanych żądań HTTP.
- Koszty – darmowy plan jest ograniczony, zaawansowane funkcje wymagają subskrypcji.
- Opóźnienia – dodatkowy skok przez sieć Cloudflare może wprowadzać niewielki narzut.
- Wymóg połączenia wychodzącego – potrzebny dostęp do portu 7844 i adresów Cloudflare.

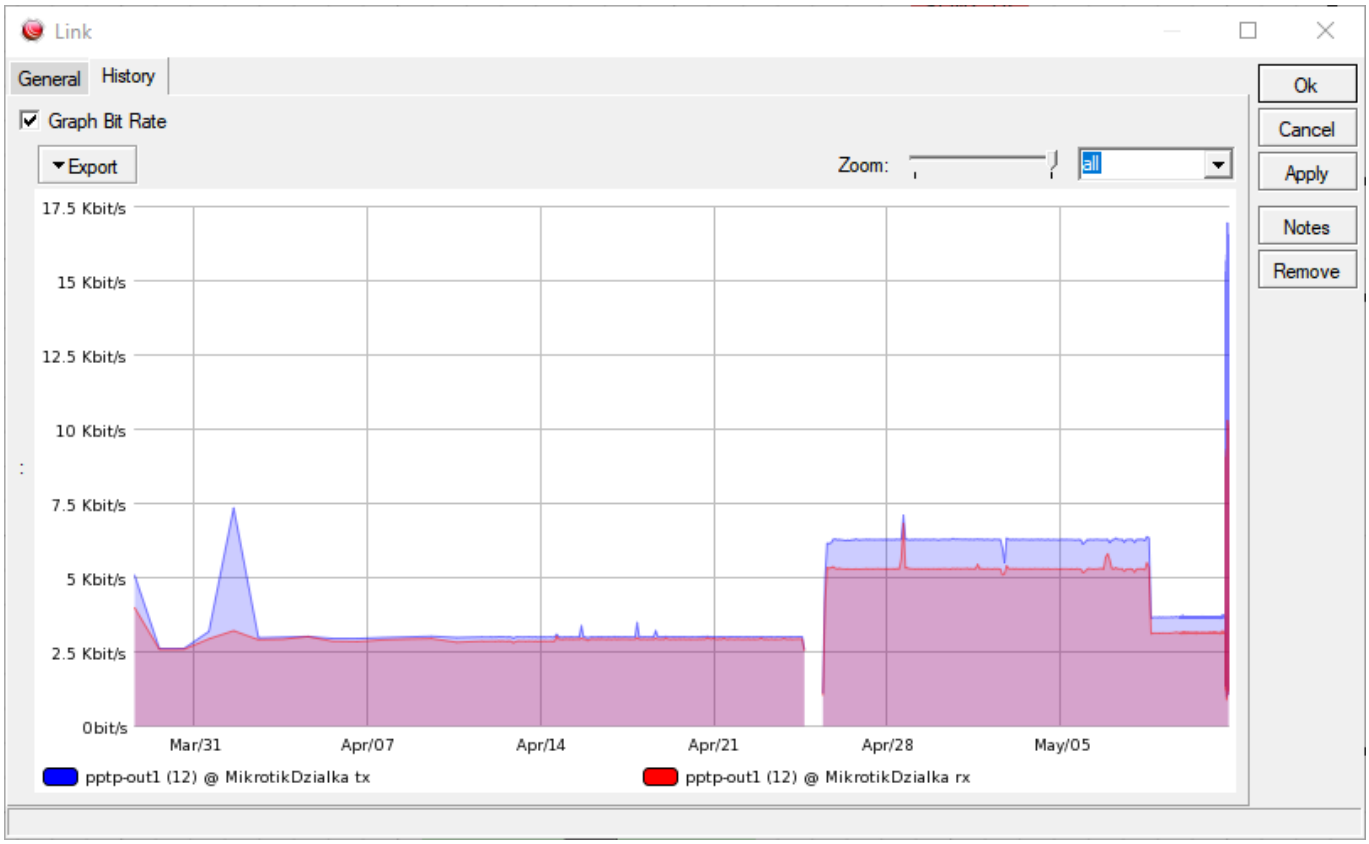
Zastosowania praktyczne

- Udostępnianie aplikacji webowych lub API – bez publicznego IP i z kontrolą dostępu.
- Zdalny dostęp do pulpitów, serwerów lub paneli administracyjnych.
- Wdrożenia w firmach z ograniczonym dostępem do Internetu lub siecią za NAT-em.
- Współdzielenie środowisk developerskich – bez ekspozycji usług na zewnątrz.

Monitorowanie mojej sieci



Nie mam na tyle kluczowych usług w mojej sieci żeby to pieczołowicie monitorować natomiast mam na przykład skonfigurowane powiadomienia mailowe w cloudflare jeżeli któryś z tuneli się rozłączy to dostaję od razu informację mailową który. Jeżeli chodzi o resztę monitorowania to wykorzystuję do tego LAB-ROUTER na którym mam uruchomioną usługę The Dude pozwala mi ona na prostą mapę sieci oraz na prosty monitoring usług takich jak Ping, HTTP, SSH itp.



Mam skonfigurowane proste monitorowanie trafficu na wszystkich linkach co pozwala mi czasami monitorować ewentualne anomalie.

Domeny

Jak można zauważyć we wcześniejszych akapitach mam dwie domeny `ardugeek.ovh` oraz `ostrowski.net.pl`. Spieszę w wyjaśnieniem, w momencie kiedy miałem mało osób które uczyłem i dawałem szkolenia to kupiłem sobie domenę do hostowania własnej strony do eksperymentów i zabawy. Z wiekiem i doświadczeniem doszedłem do wniosku że lepiej będzie mieć maila i domenę z własnym nazwiskiem więc też i taką kupiłem. Obydwie sa opłacone na następne nie pamiętam ile lat, więc nadal mam je obydwie, i przyjąłem taką konwencję że rzeczy które są bardziej permanentne takie jak to WIKI albo moja strona będą pod domeną `ostrowski.net.pl`, a rzeczy które są eksperymentami lub dla zabawy zostawiam na domenie ardugeek.ovh.

Stąd taka geneza dwóch domen. Można by powiedzieć że jedna to IntraNet a druga to ExtraNet tylko że byłoby to trochę nie poprawne wykorzystanie tych pojęć .

Podsumowanie

W tym artykule przedstawiłem projekt mojej domwej sieci (niektórzy wolą to określać mianem Home Labu), z której korzystam na co dzień słuchając muzyki na smartfonie czy sprawdzając swoje notatki na mojej WIKI. Jak można zauważyć jest to mieszanka wielu różnych usług, programów i systemów operacyjnych połączona razem w jedne wielkie (tutaj każdy sobie mówi sam co), jest to wynikiem hostowania różnych usług przez lata na rożnych starych laptopach albo komputerach z odzysku. Ta konfiguracja sieci jest ze mną już ładne parę lat i odnoszę wrażenie że jest to jej finalna wersja, od teraz już tylko spodziewam się małych zmian ale nie jej całej przebudowy.

Sieć ta też była i nadal czasami jest wykorzystywana do hostowania projektów moich znajomych lub rodziny. Swego czasu też było na niej hostowane kilka stron dla szkół które obsługiwałem. Lata dokładania technologi zmieniania oraz usuwania spowodowały że jest to jak stara dobra wytarta kanapa, jak się na niej usiądzie to czujesz się jak u siebie.