Subiektywny poradnik administratora

cz. IV

Zdjęcie wygenerowane za pomocą sztucznej inteligencji

Część dalsza uniezależniania się od korporacji z USA. Dla równowagi, następny odcinek cyklu będzie traktował o amerykańskim rozwiązaniu korporacyjnym. Wracamy więc do open source.

Adam Jurkiewicz

administrator sieci i serwerów Linux od ponad 25 lat. Programista Pythona, zwariowany nauczyciel młodzieży. Zdecydowany zwolennik oprogramowania open source i systemów Linux od 1993 r., których od ponad dwóch dekad używa w codziennej pracy. Członek zarządu Sekcji Informatyki Szkolnej przy PTI oraz członek oddziału mazowieckiego PTI. Dostępny w sieciach społecznościowych:
https://www.linkedin.com/in/adam-jurkiewicz-python-linux/
https://linux.social/@adam_jurkiewicz
https://jurkiewicz.chat

Tym razem zaprezentuję komunikator (żaden Signal czy inny WhatsApp) oraz aplikację do zdalnego przechwytywania pulpitu (ulubione narzędzie łowców naiwnych na hasło „ktoś próbuje przechwycić Twoje konto bankowe, nasz specjalista Ci dopomoże ;-)". Oprócz takich „specjalistów" są jeszcze prawdziwi pomocnicy, którzy potrzebują dostępu do naszego urządzenia.

Jako administratorzy sieci często stoimy przed dylematem – jak zapewnić sprawną komunikację dla pracowników przedsiębiorstwa? Powinna spełniać następujące wymagania:

- **szyfrowanie end2end – to w dzisiejszych czasach podstawa bezpiecznej komunikacji, najlepiej dobrym, bezpiecznym szyfrem: np. taką implementacją OpenPGP RFC 9580 oraz kombinacją Curve25519 dla aprobaty kluczy i Ed25519 dla cyfrowych podpisów;**
- **wskazanie, że druga strona odczytała wiadomość (potwierdzenie);**
- **możliwość tworzenia grup;**
- **przesyłanie wiadomości głosowych;**
- **aplikacja klienta powinna działać na maksymalnie dużej liczbie systemów operacyjnych, zarówno w wersji desktop (komputer, laptop), jak i na urządzeniach mobilnych (Android, iOS);**
- **połączenie z numerem telefonu (pytanie – czy to jest niezbędne?).**

21 października 2025 r. miała miejsce całkiem poważna awaria AWS (tekst piszę pod koniec października), co jest kolejnym argumentem na rzecz utrzymywania własnych rozwiązań, prywatnych i możliwie w odseparowanych od siebie miejscach. Co z tego, że usługa A jest wykupiona u dostawcy AD, a usługa B – u dostawcy BD, gdy obaj dostawcy wykorzystują tego samego dostawcę usług (np. AWS).

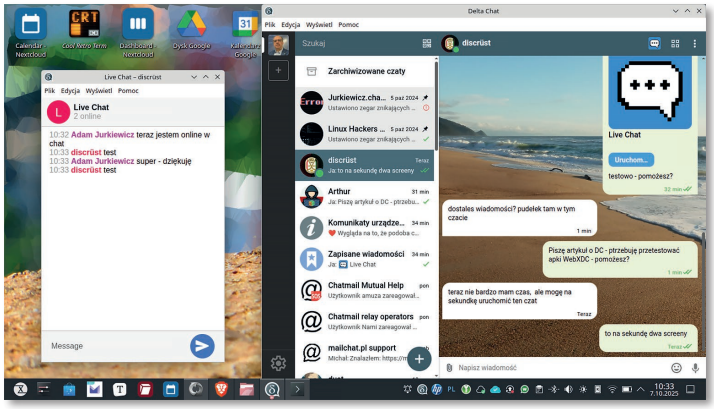
Światła na Delta Chat

Delta Chat to niezawodna, zdecentralizowana i bezpieczna aplikacja do przesyłania wiadomości, dostępna na platformy mobilne i stacjonarne.

NixFAQ pisał na swoich stronach już kilka lat temu (https://nixfaq.org/2020/09/delta-chat-a-libre-decentralized-chat-over-email-end-to-end-encrypted-messaging-solution.html):

„Delta Chat wykorzystuje istniejące adresy e-mail jako identyfikatory. Domyślnie żadne dane książki adresowej (takie jak numer telefonu) nie są odczytywane ani przesyłane do serwerów zewnętrznych, jak ma to miejsce w przypadku komunikatorów takich jak WhatsApp i spółka. W przeciwieństwie do wielu innych komunikatorów Delta Chat udostępnia identyfikator, który nie jest powiązany z numerem telefonu. Mile widziany wyjątek."

Oto screen mojej aplikacji na komputerze, której używam równolegle z telefonem – przyznacie, że ciekawy ekran. Zwracam szczególną uwagę na okno „LiveChat" – to mini aplikacja, która potrafi działać wewnątrz rozmowy lub grupowego chatu – zapewnia komunikację wewnątrz, między dwoma osobami.



Kilka słów na temat zabepieczeń. Delta Chat wykorzystuje bezpieczny podzbiór standardu OpenPGP do automatycznego szyfrowania typu end-to-end za pomocą następujących protokołów:

- Secure-Join do wymiany informacji o konfiguracji szyfrowania poprzez skanowanie kodów QR lub „linki zaproszeń";
- Autocrypt do automatycznego ustanawiania szyfrowania typu end-to-end między kontaktami a wszystkimi członkami czatu grupowego;
- Delta Chat nie wysyła zapytań, nie publikuje ani nie wchodzi w interakcję z żadnymi serwerami kluczy OpenPGP.

Jeśli chcesz wiedzieć więcej, odsyłam do serwisu Github https://github.com/deltachat/ oraz na stronę projektu https://delta.chat/pl/

Pora na własny, prywatny serwer Delta Chat

Chcę w mojej firmie wdrożyć takie rozwiązanie, by mieć kontrolę nad transportem wiadomości (bo nad ich treścią nie będę miał żadnej kontroli) – mocne szyfrowanie po stronie klienta (urządzenia) uniemożliwi mi czytanie tego, co handlowiec pisze do dyrektora oddziału czy managerka do innego handlowca. To pozostanie ich tajemnicą, zresztą, teoretycznie, wszystkie komunikatory mają tę cechę, przynajmniej według zapewnień producentów.

Zapraszam więc na stronę https://jurkiewicz.tech – to mój prywatny serwer, który jest jednak otwarty dla nowych użytkowników (ostatnio nawet dowiedziałem się, że korzystają z niego rosyjscy aktywiści ukrywający swoją komunikację przed Putinem). A potem na stronie https://github.com/chatmail/relay znajdziesz pełny opis procedury instalacyjnej. Z własnego doświadczenia podpowiem, że warto użyć Linuksa jako klienta lub WSL w przypadku Windows. Po prostu skrypty instalacyjne wykorzystują `ssh`, więc klucze są ważne, by się połączyć. O konfiguracji `ssh` i kluczy piszę na moim blogu https://blog.jurkiewicz.tech/ssh-from-linux-to-linux-in-10-seconds-35b46ffd31cd.

Zakładam, że dla celów tego komunikatora masz swojego VPS (podobnie jak do NextCloud) i domenę. Koszty roczne nie przekraczają kilkuset złotych (piszę z własnego doświadczenia). Jeśli potrzebujesz pomocy, proponuję wizytę w serwisie https://support.delta.chat/ – to świetne wsparcie zaangażowanej społeczności. Powodzenia – a gdy już będziesz po wdrożeniu, skontaktuj się ze mną (uwaga! link działa tylko wtedy, gdy masz już aplikację Delta Chat na urządzeniu, z którego go wywołujesz):

https://i.delta.chat/#0E69B1588366C3D777D05B982634A08B5A31F2BC&a=dpd5gcipf%40deltachat.jurkiewicz.chat&n=Adam%20Jurkiewicz&i=-J4V_DXahxZ&s=SLr210JLHvM

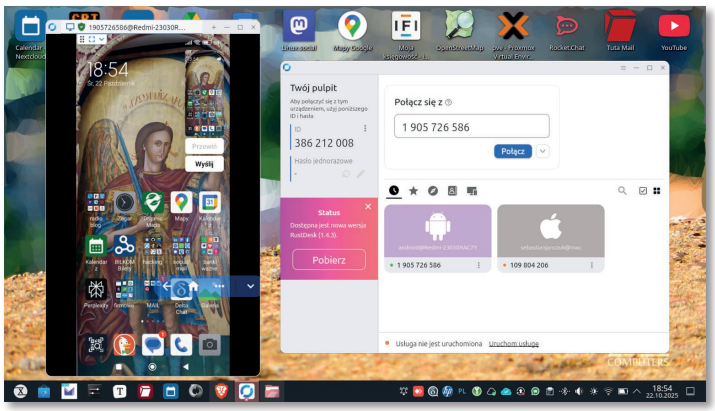
RustDesk – alternatywa dla TeamViewer czy AnyDesk...

Powiecie, że to nic nowego, przecież jest tyle innych, komercyjnych rozwiązań. To prawda, ale możemy mieć własny sewer pośredniczący, więc mamy 99% pewności, że nikt nie podejrzy naszego ekranu, nie zaloguje informacji, kto z kim i kiedy wykonał połączenie. Własny serwer to pełna kontrola nad danymi.

Możliwość samodzielnego hostowania serwera RustDesk stanowi kluczową zaletę z perspektywy RODO. Własny serwer składa się z dwóch komponentów: serwera ID (hbbs), działającego na portach TCP 21115-21116, 21118 i UDP 21116, oraz serwera przekaźnikowego (hbbr) na portach TCP 21117 i 21119. Oczywiście to wszystko może być zainstalowane na kontenerze Linux na naszym Proxmox, którego posiadamy w jakieś serwerowni lub na VPS.

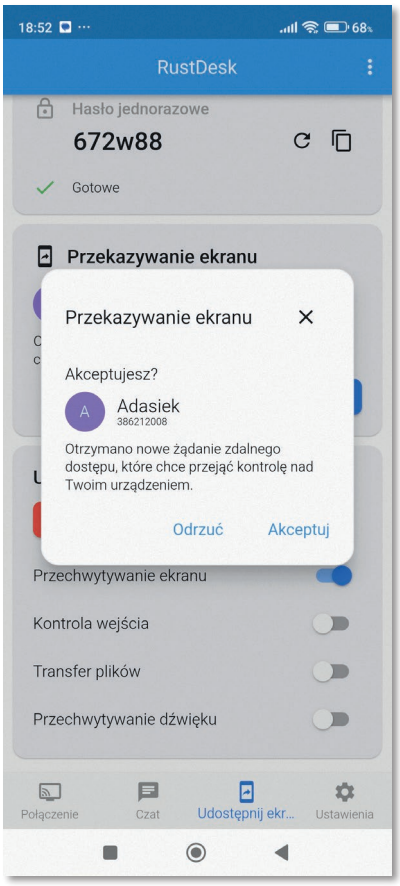
Więcej o tym rozwiązaniu (w tym filmy na YouTube) przeczytasz na moim blogu: https://blog.jurkiewicz.tech/rustdesk-an-opensource-alternative-to-teamviewer-or-anydesk-c1e058bd5f0f. Na stronie: https://github.com/rustdesk/rustdesk/releases znajdziesz aplikacje dla każdego systemu operacyjnego.

Popatrzcie na kilka ekranów połączenia z komputera do mojego telefonu z systemem Android (RustDesk jest również dostępny dla iOS).

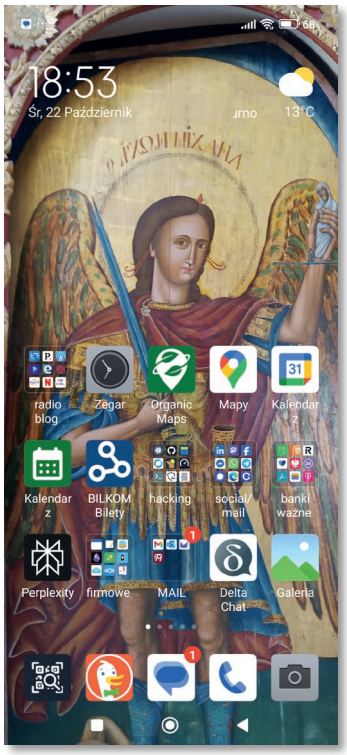


To jest wykonane połączenie – aby je wykonać, muszę uruchomić współdzielenie ekranu w telefonie – to nie jest łatwe, bowiem aplikacja informuje, że być może nie wiem, co robię, choć ja doskonale wiem.

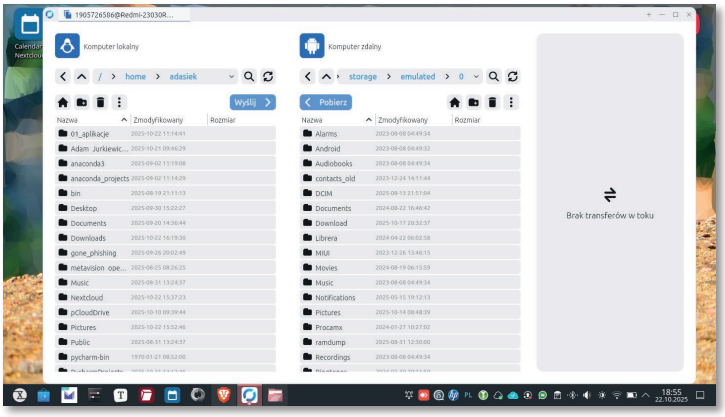
Przychodzące połączenie muszę dodatkowo zaakceptować:



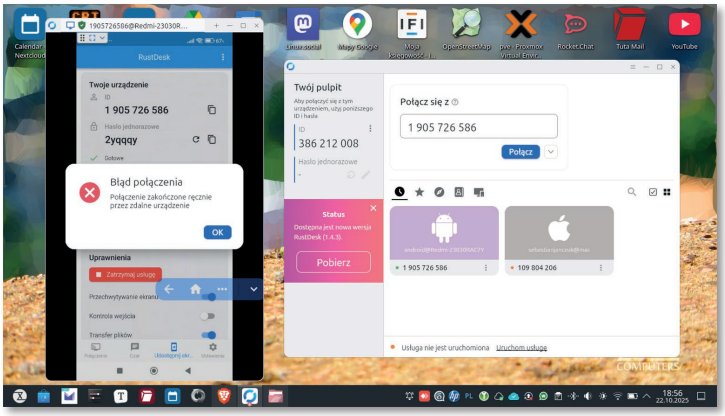
Dopiero wtedy uzyskuję na komputerze dostęp do urządzenia, mogę nim nawet sterować. Poniżej zrzut ekranu telefonu, by było widać, że wykonałem je w tym samym czasie, co zrzut ekranu komputera.



Dodatkowo mogę również kopiować pliki pomiędzy urządzeniami.



Jeśli telefon rozłączy się, na komputerze zobaczę, że już nie mam dostępu do urządzenia.



Uzyskujemy:

- **pełną kontrolę nad danymi – wszystkie informacje pozostają w infrastrukturze organizacji;**
- **brak zewnętrznych serwerów – eliminacja konieczności przesyłania danych przez serwery trzecich stron;**
- **własne zasady przetwarzania – możliwość ustalenia własnych procedur zgodnych z RODO;**
- **transparentność – pełną wiedzę, gdzie i jak dane są przechowywane.**

RustDesk wykorzystuje szyfrowanie end-to-end bazujące na bibliotece NaCl (tej samej co Signal). Wszystkie połączenia są domyślnie szyfrowane, nawet przy korzystaniu z publicznych serwerów. Własny serwer pozwala na dodatkową konfigurację ustawień szyfrowania.

Zaprezentowałem dwie kolejne aplikacje, które pozwalają nam na niezależność – może nie są tak bardzo znane, jak Signal czy TeamViewer, ale na pewno legalnie dostępne i bezpłatne dla każdego, nawet dla korporacji. I co istotniejsze, dane pozostają naszą własnością i są pod naszą kontrolą, a nie korporacji X czy Y... to ważne w dzisiejszych czasach.