NEURO-SYMBOLIC AI: KOOPERACJA LOGIKI Z SIECIAMI NEURONOWYMI

Index: 285358
www.programistamag.pl
Magazyn programistów i liderów zespołów IT

programista

4/2026 (125)
Cena 33.90 zł (w tym VAT 8%)

ESP-NOW

Protokół komunikacyjny dla drona

JUŻ W EMPIKU



ISSN 2084-9400 04

- Komputer niebinarny. Czy to jest możliwe?
- Programista wspomagany sprzętowo
- Czy można zastąpić płatne AI lokalnymi?
- Własny bot na Discordzie
- ECS w silnikach gier i symulacji
- Strefa CTF

PERFECTION IN PROTECTION LICENSING SECURITY
WIBU SYSTEMS

Zaszyfruj. Dostarcz. Uruchom – bezpieczna aktualizacja firmware w dronach

Dron kojarzy się przede wszystkim z lataniem. Z punktu widzenia producenta jest również komputerem, który przez cały okres eksploatacji trzeba bezpiecznie aktualizować. Jego możliwości zależą w dużej mierze od oprogramowania pokładowego.

Aktualizacja firmware pozwala usuwać błędy, zamykać podatności i dodawać funkcje, ale jest to zarazem z punktu widzenia bezpieczeństwa wrażliwy etap cyklu życia urządzenia. Jeżeli atakujący podmieni pakiet, skopiuje go albo uruchomi na nieautoryzowanym urządzeniu, skutki mogą wykraczać poza utratę własności intelektualnej. Zmodyfikowany kod może zmienić sposób działania systemu lub ominąć ograniczenia producenta.

Dlatego proces aktualizacji powinien odpowiadać na trzy pytania: czy pakiet pochodzi od producenta, czy nie został w międzyczasie zmieniony oraz czy dane urządzenie ma prawo go uruchomić. Okazuje się, że rozwiązanie można oprzeć na znanym mechanizmie wykorzystywanym w licencjonowaniu: kryptograficznej ochronie firmware i kluczu przechowywanym w kontenerach licencyjnych. W dalszej części przedstawiam wykorzystanie komponentów frameworka CodeMeter do ochrony firmware.

Firmware jako chronione oprogramowanie

Pierwszy etap procesu polega na zabezpieczeniu firmware i ma miejsce po stronie producenta. Plik wykonywalny lub wybrane moduły są zabezpieczane za pomocą AxProtector, który szyfruje kod i dodaje kontrolę integralności. Klucze generowane są m.in w oparciu o Firm Code (identyfikator producenta) i Product Code (identyfikator produktu), a następnie zapisywane w postaci licencji w kontenerze licencyjnym. Przechwycenie aktualizacji nie daje dostępu do jej jawnej postaci ani możliwości uruchomienia jej bez odpowiedniego uprawnienia, tj. licencji.

Jeżeli firmware jest aplikacją natywną działającą pod kontrolą Embedded Linux, można zastosować dedykowany AxProtector, który automatycznie dodaje funkcje zabezpieczające, takie jak szyfrowanie, zaawansowane zaciemnianie kodu czy przenoszenie kodu do bezpiecznego modułu, np. układu CodeMeter ASIC. Chroniony kod lub jego fragment pozostaje zaszyfrowany do chwili, w której jest potrzebny.

Jeżeli firmware działa bezpośrednio na mikrokontrolerze, bez warstwy systemu operacyjnego (ang. *bare metal*) albo pod kontrolą wyspecjalizowanego systemu RTOS, CodeMeter Embedded – dostępny również w postaci kodu źródłowego – można zintegrować z firmware lub z dodatkowym, odpowiednio dostosowanym boot loaderem. W ten sposób można chronić zarówno proces aktualizacji firmware, jak i jego późniejsze wykonanie.

Kontener jako kotwica zaufania

Gdzie przechowywać klucze i uprawnienia potrzebne do uruchomienia firmware? Umieszczenie ich na stałe w kodzie przenosi problem do pliku wykonywalnego. Atakujący może analizować aplikację, poszukiwać klucza w pamięci i odtwarzać proces deszyfrowania.

Funkcję kotwicy zaufania pełni kontener licencyjny – CmContainer. To na nim zapisana jest licencja oraz chronione dane używane w operacjach kryptograficznych. Framework CodeMeter obsługuje różne typy kontenerów. CmActLicense jest zaszyfrowanym i podpisanym plikiem związanym z cyfrowym odciskiem urządzenia. CmReady wiąże kontener z kartą pamięci. Gdy potrzebna jest ochrona sprzętowa, można zastosować CmDongle (token sprzętowy) lub układ CmASIC, zintegrowany z elektroniką urządzenia. Wybór zależy od platformy, interfejsów i wymaganej odporności na analizę fizyczną.

Weryfikacja na urządzeniu

Po dostarczeniu do drona pakietu z aktualizacją mechanizm ochrony uzyskuje dostęp do wpisu licencyjnego w konterze. Na wydajnym komputerze pokładowym może za to odpowiadać serwis CodeMeter Runtime. Dla urządzeń o mniejszych zasobach przeznaczony jest CodeMeter Embedded. Biblioteka ma, zależnie od platformy, około 400–600 KB i udostępnia podzbiór CodeMeter Core API.

Odszyfrowawnie aplikacji ma miejsce każdorazowo podczas jej uruchamiania. Brak właściwego kontenera, licencji lub integralności powoduje odrzucenie pakietu albo uniemożliwia uruchomienie kodu.

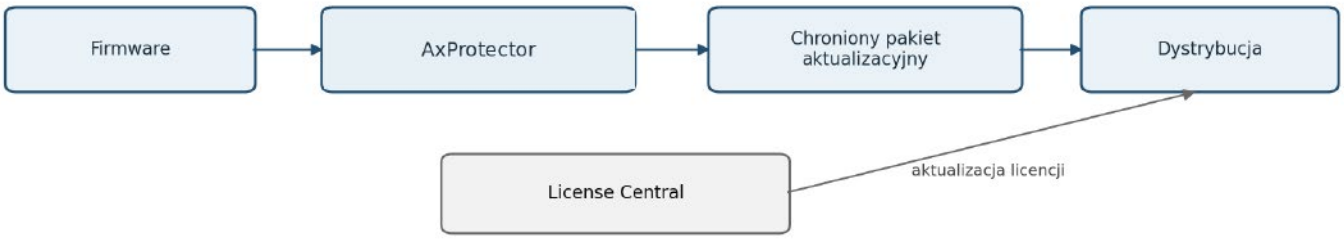
Jedna aktualizacja dla wielu urządzeń

Powiązanie ochrony z licencją oddziela dystrybucję pliku od nadawania prawa do użycia. Producent może udostępnić jeden zaszyfrowany pakiet całej flocie, a uprawnienie przekazać tylko określonym urządzeniom. Product Code może reprezentować produkt, wariant urządzenia lub chronioną funkcję. Parametry licencji mogą ograniczać dostęp w czasie albo uzależnić go od okresu utrzymania.

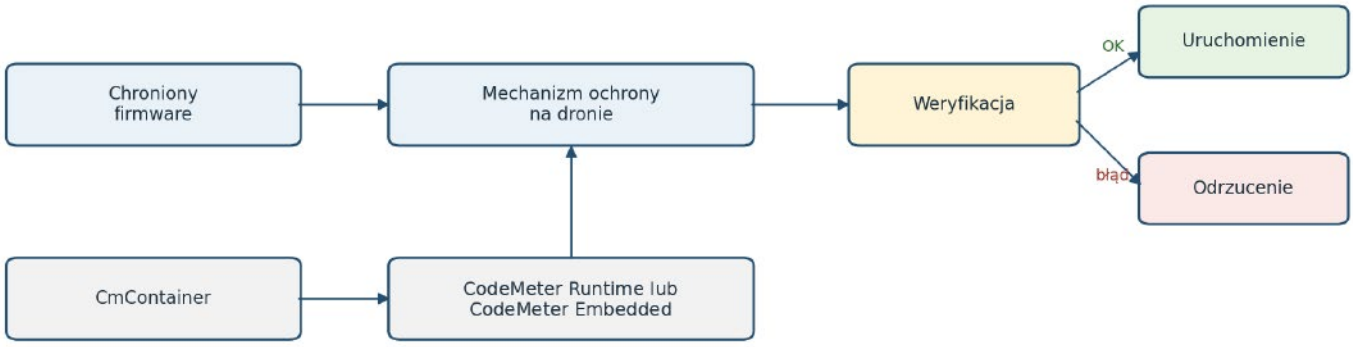
Procesem może zarządzać CodeMeter License Central. System tworzy i dystrybuuje licencje oraz ich aktualizacje, ale nie uczestniczy w lokalnym odszyfrowywaniu firmware. Urządzenie może otrzymać uprawnienie online albo przez plikową aktualizację kontenera, co ma znaczenie w odizolowanych sieciach. Licencja i firmware pozostają osobnymi artefaktami, choć mogą być elementami jednej procedury serwisowej.

Ochrona całego cyklu aktualizacji

Połączenie ochrony (AxProtector), mechanizmu wykonawczego (CodeMeter Runtime / CmEmbedded) oraz kontenera licencyjnego tworzy jeden spójny proces. Producent zabezpiecza jeden pakiet i zachowuje kontrolę nad prawem do jego uruchomienia. Kontener



Rysunek 1. Przygotowanie chronionej aktualizacji i uprawnienia



Rysunek 2. Weryfikacja firmware na urządzeniu docelowym

przechowuje klucze, a License Central może zarządzać uprawnieniami w cyklu życia floty dronów.

Nie każda platforma dronowa obsłuży identyczny wariant tej architektury. Wydajny komputer z Embedded Linux ma inne możliwości niż kontroler czasu rzeczywistego z ograniczoną pamięcią. Niezmienna pozostaje jednak zasada: firmware powinien pozostać chroniony do chwili, w której autoryzowane urządzenie potwierdzi właściwą licencję, integralność pakietu i prawo do uruchomienia danej wersji.

CodeMeter obejmuje zestaw narzędzi przeznaczonych dla różnych platform sprzętowych i programowych – od niewielkich mikrokontrolerów po wydajne stacje robocze i serwery. Niezależnie od platformy wykorzystuje ten sam interfejs API, format licencji, koncepcję kontenerów oraz mechanizmy zarządzania licencjami.

Ale CodeMeter oferuje znacznie więcej niż tylko ochronę procesu aktualizacji firmware. Pozwala bezpiecznie udostępniać dodatkowe funkcje, ograniczać czas korzystania z nich do określonego okresu oraz wycofywać wcześniej przyznane licencje.



JANUSZ HRYSZKIEWICZ

janusz.hryszkiewicz@wibu.com

Związany z IT od 25 lat, głównie w zakresie rozwiązań dedykowanych. Autor kilku rozwiązań dla biznesu, które stały się produktami. Obecnie zaangażowany w zwiększenie cyberodporności oprogramowania, AI oraz systemów wbudowanych.

/* REKLAMA */

WIBU SYSTEMS

Licensing-as-a-Service dla producentów oprogramowania, którzy działają szybko.

Publikuj, chroń i monetyzuj swoje oprogramowanie dzięki gotowej do użycia infrastrukturze licencjonowania i bezpieczeństwa.

- Ochrona przed inżynierią wsteczną
- Gotowe modele licencjonowania
- Centralne zarządzanie licencjami, użytkownikami i dostępem
- Samoobsługowy portal dla klientów i partnerów
- Bezpieczna i skalowalna chmura

Szybszy time-to-market. Pełna kontrola nad swoim cyfrowym IP.



Dowiedz się więcej

sales@wibu.com
www.wibu.com

CodeMeter Licensing-as-a-Service