Embedded i ochrona integralności oprogramowania

Systemy sterowania są coraz częściej połączone ze sobą i komunikują się za pośrednictwem publicznych sieci. Łączą świat analogowy z fizycznym oraz cyfrowy ze światem wirtualnym. Przemysł 4.0 niesie ze sobą nie tylko wygodę i nowe możliwości, lecz także otwiera cyber-fizyczne systemy na świat zewnętrzny, zwiększając ryzyko ataków pochodzących z zewnątrz.

To dodatkowe ryzyko jest realne i wymaga wdrożenia skutecznych środków zaradczych – nie tylko w celu ochrony własności intelektualnej, lecz również zapobiegania wprowadzaniu złośliwego oprogramowania poprzez manipulację kodem. Zastosowanie odpowiednich mechanizmów ochrony integralności pozwala dodać kolejne warstwy bezpieczeństwa, które realizują te nowe zadania ochronne.

Czym jest ochrona integralności?

Pojęcie **ochrony integralności** obejmuje zestaw środków bezpieczeństwa mających na celu ochronę zasobów systemowych, programów i danych przed nieautoryzowaną manipulacją – lub co najmniej umożliwiających wykrycie i zgłoszenie takich zmian.

Wyzwanie polega na zapewnieniu nienaruszalności danych, a jeśli nie jest to możliwe – na przełączeniu systemu w tryb bezpieczny i zatrzymaniu wykonywania wszystkich funkcji.

Najlepsze rozwiązania w zakresie ochrony integralności opierają się na kryptografii oraz powiązanych z nią mechanizmach bezpieczeństwa, takich jak podpisy cyfrowe i uwierzytelnianie komunikatów.

Co łączy ochronę przed kopiowaniem z ochroną know-how?

Celem ochrony przed kopiowaniem jest zapobieganie fałszowaniu całych urządzeń. Z kolei ochrona własności intelektualnej oznacza zabezpieczenie firmowych algorytmów i metod przed inżynierią wsteczną.

Nowoczesne rozwiązania oparte na silnym szyfrowaniu i wykorzystaniu bezpiecznych elementów sprzętowych (dedykowane układy ASIC czy chipy typu smart card) mogą być wykorzystywane zarówno do ochrony oprogramowania i własności intelektualnej, jak i zapewnienia integralności systemów.

Ataki na systemy cyber-fizyczne

Poniżej przedstawiono przykłady potencjalnych ataków na systemy wbudowane:

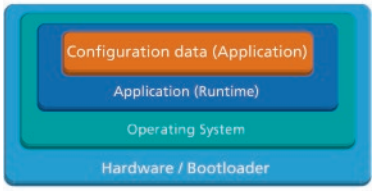
1. **Fałszywe urządzenie** – atakujący tworzą podróbkę oryginalnego urządzenia, która z zewnątrz wygląda identycznie, lecz jej funkcje zostały zmodyfikowane w celach przestępczych. Takie urządzenie może zostać zainstalowane np. jako część zamienna podczas serwisowania sprzętu.
2. **Podmiana oprogramowania** – atakujący opracowują własne oprogramowanie i uruchamiają je poprzez wymianę karty pamięci w systemie wbudowanym.
3. **Manipulacja kartą pamięci** – atakujący wyjmują kartę pamięci z systemu wbudowanego, modyfikują zapisane na niej oprogramowanie, a następnie ponownie umieszczają kartę w urządzeniu.
4. **Modyfikacja przez interfejsy komunikacyjne** – atakujący ingerują w oprogramowanie systemu wbudowanego, przejmując kontrolę nad jego interfejsami komunikacyjnymi z zewnątrz.
5. **Analiza podczas pracy systemu** – atakujący obserwują działający system wbudowany w trakcie jego normalnej pracy, aby analizować jego zachowanie i opracować możliwe wektory ataku.

Projektowanie systemu wbudowanego i związane z tym wyzwania

Zasadniczo typowy system wbudowany buduje się warstwowo. Należy przy tym pamiętać, że warstwa zewnętrzna może uzyskać dostęp do pamięci warstwy wewnętrznej, natomiast w większości przypadków warstwa wewnętrzna nie ma dostępu do warstwy zewnętrznej.

Zewnętrzna warstwa (sprzęt/bootloader) jest warstwą startową całego procesu.

1. Bootloader weryfikuje integralność systemu operacyjnego i – po pomyślnej walidacji – ładuje go.
2. System operacyjny uruchamia się dopiero wtedy, gdy bootloader został uznany za godny zaufania na podstawie wstecznej weryfikacji.



Rysunek 1. Podstawowa struktura systemu wbudowanego

3. System operacyjny weryfikuje integralność aplikacji i ładuje ją tylko wtedy, gdy pomyślnie przejdzie proces walidacji.
4. Aplikacja uruchamia się wyłącznie wtedy, gdy system operacyjny został uznany za godny zaufania w wyniku wstecznej weryfikacji. Dodatkowo aplikacja sprawdza integralność danych konfiguracyjnych i korzysta z nich tylko po ich pozytywnej weryfikacji.
5. Jeśli dane konfiguracyjne zawierają również kod wykonywalny, możliwe jest także zweryfikowanie poziomu zaufania aplikacji, zanim taki kod zostanie uruchomiony.

Implementacja mechanizmu sprawdzania integralności

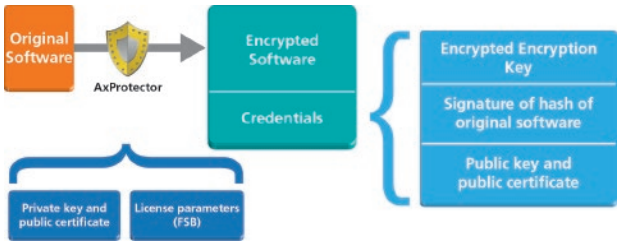
W pierwszej kolejności niezabezpieczone oryginalne oprogramowanie musi zostać podpisane cyfrowo i zaszyfrowane. W tym celu można skorzystać z AxProtector – narzędzia służącego do ochrony oprogramowania. Realizowane są następujące kroki:

1. Obliczenie wartości skrótu (hash) oryginalnego oprogramowania.
2. Podpisanie wartości skrótu przy użyciu klucza prywatnego producenta.
3. Szyfrowanie oryginalnego oprogramowania z użyciem klucza generowanego na podstawie:
   - wartości początkowej (seed value) zawartej w oryginalnym oprogramowaniu,

» tajnego klucza producenta,
» oraz innych parametrów zdefiniowanych przez wydawcę.

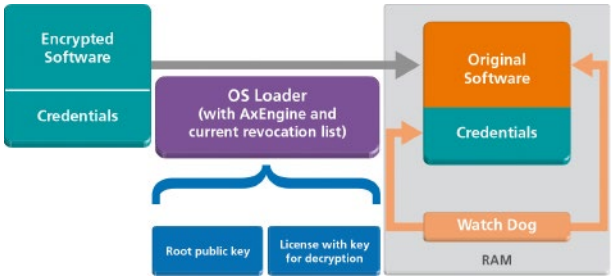
4. Dołączenie publicznej części certyfikatu podpisu do zaszyfrowanego oprogramowania.

Pierwszy etap sprawdzania integralności (tzw. weryfikacja w przód, forward check), czyli kontrola autentyczności i integralności oprogramowania lub odpowiadających mu danych przed ich załadowaniem, przebiega w następujący sposób:



Rysunek 2. Szyfrowanie i podpisywanie aplikacji

Z kolei drugi etap – **weryfikacja**, realizowany jest za pomocą zintegrowanego narzędzia systemowego – AxEngine – obejmuje następujące kroki, wykonywane w trakcie ładowania aplikacji:



Rysunek 3. Kontrola integralności Forward-Check

1. Jeśli dostępna jest ważna licencja, zaszyfrowane oprogramowanie zostaje odszyfrowane.
2. 2. Certyfikat dołączony do poświadczeń lub łańcucha certyfikatów jest weryfikowany względem publicznego klucza głównego (root key).
3. Następnie obliczana jest wartość skrótu (hash) odszyfrowanego oryginalnego oprogramowania.
4. Podpis cyfrowy tej wartości skrótu jest sprawdzany przy użyciu klucza publicznego.

Oprócz tego można wdrożyć dodatkowe środki zwiększające poziom bezpieczeństwa, takie jak zaawansowane zarządzanie certyfikatami umożliwiające autoryzowane uruchamianie oprogramowania wyłącznie na określonych urządzeniach.

Możliwe jest również zastosowanie sprawdzenia daty ważności certyfikatu lub weryfikacji listy unieważnionych certyfikatów (CRL – Certificate Revocation List). Takie kontrole mogą być dodatkowo wykonywane okresowo w czasie działania systemu (watchdog), w celu bieżącego monitorowania integralności i autentyczności środowiska wykonawczego.

Rozwiązanie oparte na technologii CodeMeter obejmuje następujące etapy, które mogą być szczególnie skuteczne również w zautomatyzowanych procesach kompilacji (build process):

» Szyfrowanie kodu programu – w celu zapobiegania analizie statycznej kodu oraz inżynierii wstecznej.
» Podpisywanie kodu programu – obejmujące zarówno aplikację, jak i obraz systemu operacyjnego.
» Przechowywanie współdzielonych poufnych informacji wykorzystywanych do deszyfrowania.
» Przechowywanie klucza prywatnego podpisu wyłącznie po stronie producenta.
» Weryfikacja podpisu i wartości hashujacych podczas ładowania oraz w trakcie działania systemu (runtime).

Podsumowanie

Wibu-Systems, dzięki technologii CodeMeter, oferuje oparty na karcie kryptograficznej system ochrony, który dostępny jest dla stosowanych w przemyśle interfejsów. CodeMeter obsługuje popularne systemy operacyjne, takie jak Windows, macOS, Linux, a także Windows Embedded, Real Time Linux, VxWorks oraz sterowniki PLC, w tym CODESYS i inne. System posiada bezpieczne implementacje metod szyfrowania symetrycznego i asymetrycznego (AES, RSA, ECC), funkcje skrótu (SHA-256), funkcje weryfikacji podpisów cyfrowych (ECDSA) oraz generator liczb losowych.

JANUSZ HRYSZKIEWICZ

janusz.hryszkiewicz@wibu.com

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

WIBU SYSTEMS

CodeMeter – Od kodu do sukcesu

Generuj realne przychody swoim oprogramowaniem dzięki CodeMeter.

- **Wszechstronna monetyzacja:** Licencjonowanie dostosowane do wymagań rynku.
- **Skuteczna ochrona IP:** Innowacyjne szyfrowanie i ochrona integralności.
- **Pełna kompatybilność:** Bezproblemowa integracja z wieloma platformami.
- **Przyszłościowe rozwiązania:** Zaprojektowane, by rosły wraz z Twoimi wymaganiami.

CodeMeter to solidne podstawy i nowe możliwości dla Twojego oprogramowania.

sales@wibu.com
www.wibu.com

Rozpocznij teraz i zamów CodeMeter SDK
wibu.com/pl/sdk