https://www.elektroniknet.de/elektronik/embedded/keine-chance-fuer-hacker-171928.html

elektroniknet.de

Embedded-Software verschlüsseln

Keine Chance für Hacker

05.12.2019

Tobias Schlichtmeier

Die Tools von Wibu Systems verhindern ein Hacken der eingesetzten Software eines Embedded-Systems.

Embedded-Systeme abzusichern ist für Entwickler oft schwierig. Doch wer möchte schon, dass seine gerade entworfene Software geknackt wird? Wibu-Systems verhindert das mit eigens dafür entwickelten Tools.

Nicht alleine private PC-Anwender müssen ihre verwendete Software schützen. Auch Entwickler von Embedded-Systemen brauchen einen wirkungsvollen Schutz ihres Systems. Zum einen, um den Nachbau von Geräten zu erschweren, zum anderen um das Wissen, das mehr und mehr in der Software von Embedded-Systemen steckt, vor Reverse-Engineering zu schützen. Außerdem müssen Entwickler heutzutage **neue Geschäftsmodelle** eröffnen und die Integrität des Gesamtsystems gewährleisten.

Im Bewusstsein von Anwendern und Geräteherstellern ist das Thema **Sicherheit von Embedded-Systemen** stärker präsent als noch vor einiger Zeit. Regierungen vieler Länder haben in der Zwischenzeit »Cyber-Abwehrzentren« gegründet, um kritische Infrastrukturen wie Verkehrsleitsysteme, Energie- sowie Wasserversorgungsysteme vor Hacker- und terroristischen Angriffen zu schützen. Dabei gewinnt der Aspekt der unerlaubten Manipulation zunehmend an Bedeutung. Ein Beispiel: Hersteller von Windkraftanlagen sind bestrebt zu verhindern, dass deren Betreiber etwa über Tuning mehr Energie als vorgesehen mit der Windkraftanlage erzeugen. Denn das bedeutet mehr Verschleiß und geht damit zu Lasten der Lebensdauer. Somit erhöhen sich die Kosten der Gewährleistung für den Hersteller.

https://www.elektroniknet.de/elektronik/embedded/keine-chance-fuer-hacker-171928.html

Verschlüsseln der Software

Präventive Schutzsysteme erschweren den Nachbau von Embedded-Systemen, Geräten, Steuerungen, Maschinen und Anlagen. Sie schützen die Software mithilfe einer **Verschlüsselung des Programm-Codes** und der zugehörigen Daten. Die sichere Schlüsselspeicherung erfolgt in einem sicheren Baustein wie einem Hardware-Dongle oder einer softwarebasierten Aktivierungsdatei mit individueller Bindung an einen Fingerabdruck des jeweiligen Gerätes.

Der IP-Schutz (Intellectual Property), also der Schutz vor Reverse Engineering, wird erreicht, indem der Programm-Code verschlüsselt im **Sekundärspeicher des Zielsystems** liegt und somit eine statische Analyse per Disassemblieren unmöglich ist. Darüber hinaus sind Mechanismen eingebaut, die einen »Angriff« erkennen und den damit verbunden Schlüssel sofort sperren. Somit ist das Entschlüsseln des Programm-Codes anschließend nicht mehr möglich.

Um eine Manipulation – also ein unberechtigtes Verändern des Programm-Codes – zu erkennen, wird der **Code vom Hersteller • »digital signiert«**.
Der Schutzmechanismus auf dem Embedded-System erlaubt es, ausschließlich korrekt signierte Programme zu laden und zu installieren. Eine manipulierte Embedded-Software ist dann nicht mehr ausführbar.

CodeMeter für Embedded-Software

Zum Schutz von PC- und Embedded-Software entwickelt Wibu-Systems die **Anwendung »CodeMeter«** seit mehr als 18 Jahren kontinuierlich weiter. Mithilfe des Schutzes möchte Wibu die unterschiedlichen Bedürfnisse von Herstellern und Anwendern erfüllen.

Das Herzstück der CodeMeter-Technik ist dabei die Speicherung **kryptografischer Schlüssel und Lizenzbedingungen** (wie Pay-per-Use-Zähler, zeitliche Begrenzungen) in sicheren, nicht kopierbaren Containern. Das Spektrum reicht dabei vom Cloud-Speicher über rein softwarebasierte Aktivierungsdateien bis hin zu verschiedenen Bauformen sicherer Hardware-Elemente.

Ergänzt wird der Kern der Technik mit der **»CodeMeter Protection Suite«**, die es dem Softwarehersteller ermöglicht, seinen Programm-Code zu verschlüsseln und Integritäts- sowie Authentizitätsprüfungen zu realisieren. Dazu gehören unter anderem automatische Debugger-Erkennungen, digitale Signaturen und deren Prüfung zur Lade- und Laufzeit. Wird ein Debugger erkannt, beendet sich die geschützte Embedded-Software sofort oder wird erst gar nicht gestartet. So verhindert die automatische Debugger-Erkennung die dynamische Analyse von geladenem Programm-Code. Zusätzlich unterstützt die CodeMeter Protection Suite den automatischen Schutz der gesamten Embedded-Software. Des Weiteren stellt das Tool den individuellen Schutz einzelner Code-Abschnitte für native Programm-Codes (Maschinencode) wie ».NET MSIL« und »Java Byte«-Code sicher.

Die CodeMeter-Core- Programmierschnittstelle

Daneben steht dem Softwarehersteller die CodeMeter-Core-Programmierschnittstelle zur Verfügung, die ihm den vollen Zugriff auf alle CodeMeter-Funktionen ermöglicht. Sie kann für individuelle Anwendungsfälle auf die kryptografischen Funktionen der CodeMeter-Technik zurückgreifen und so eine **eigene gesicherte Kommunikationsschnittstelle** umsetzen. Dazu kann ein Hersteller auf die in

CodeMeter verfügbaren asymmetrischen kryptografischen Algorithmen wie **ECC** (Elliptic Curve Cryptography) und **RSA** (Rivest Shamir Adleman) zurückgreifen, um einen privaten Schlüssel auszutauschen. Anschließend kann er den Schlüssel für eine symmetrische Verschlüsselung mittels **AES** (Advanced Encryption Standard) zur sicheren Datenübertragung einsetzen. Für die CodeMeter-Core-Programmierschnittstelle steht dem Softwarehersteller eine komfortable, interaktiv nutzbare Schnittstellenbeschreibung zur Verfügung. Er kann darin – ohne dies sofort in den eigenen Quellcode einzubauen – in der Hilfe testen und simulieren und sich für seine Entwicklungsumgebung Code-Blöcke zur späteren Integration in den eigenen Quellcode erzeugen lassen. Anschließend kann er die Blöcke in die eigene Embedded-Software übernehmen **(Bild 3)**.

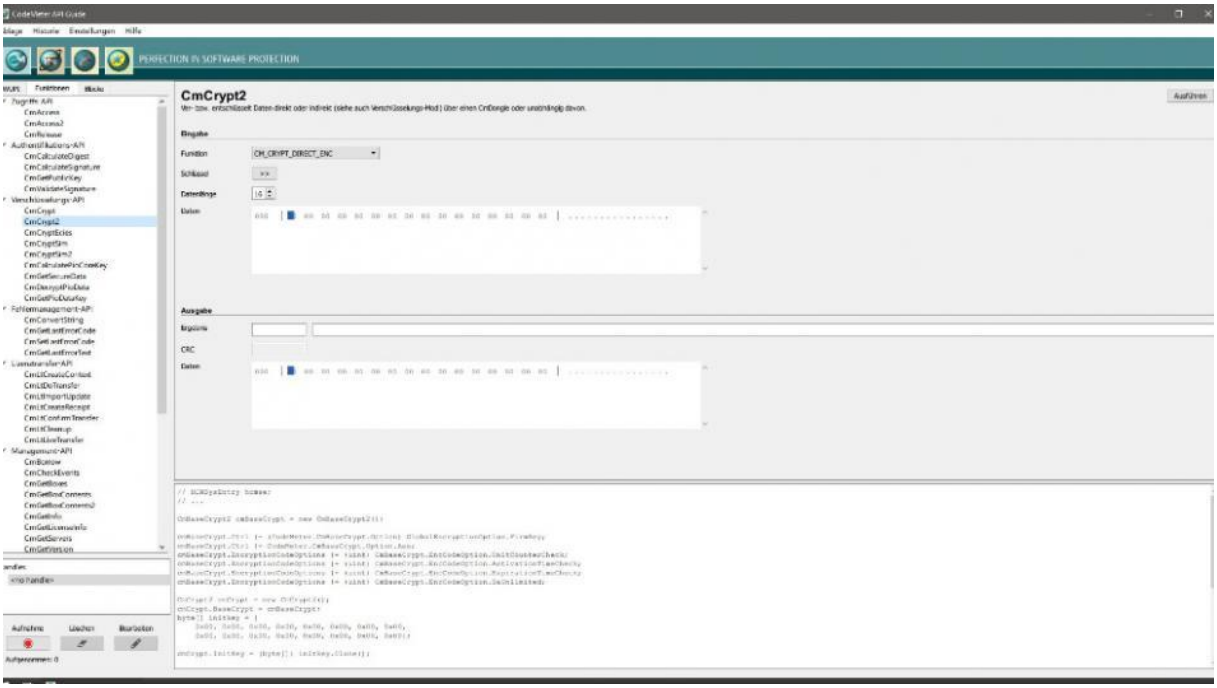

Bild 3. Die Hilfe für die CodeMeter-Core-Programmierschnittstelle.

IP- und Kopierschutz mit individueller Lizenzierung

Für den Fall, dass zusätzlich zum reinen IP-Schutz eine **individuelle Lizenzierung** einschließlich Kopierschutz zum Einsatz kommen soll, ist eine der beiden verfügbaren CodeMeter-Laufzeitumgebungen erforderlich.

Für leistungsfähige Embedded-Systeme, die 64- oder 86-bit-Hardware verwenden, steht für die Betriebssysteme Windows, Windows Embedded, Linux, Embedded Linux und macOS die vollständige Standardlaufzeit zur Verfügung. In dem Fall wird auf dem jeweiligen Betriebssystem ein Dienst beziehungsweise ein **»Daemon«** installiert. Er erlaubt es, jeden beliebigen der verfügbaren sicheren, nicht kopierbaren Container zur Speicherung der Schlüssel und Lizenzinformationen zu verwenden.

Falls das Embedded-System keine Standardhardware wie »Arm«, »PowerPC« oder »MIPS« besitzt beziehungsweise Embedded-Betriebssysteme wie »QNX«, »VxWorks«, »Android« oder »Embedded Arm Linux« verwendet, kann alternativ die **CodeMeter Embedded-Laufzeitumgebung** zum Einsatz kommen. In dem Fall wird eine statische Bibliothek in die Embedded-Software eingebunden und auf die Installation eines Dienstes oder Daemons verzichtet. Für die eben aufgelistete Hardware und Betriebssysteme stehen dafür bereits vorgefertigte Bibliotheken bereit.

https://www.elektroniknet.de/elektronik/embedded/keine-chance-fuer-hacker-171928.html

Soll ein weniger gängiges, proprietäres Betriebssystem oder sogar ein sogenanntes »Bare Metal«-Embedded-System unterstützt werden, ist der »ANSI C« Quell-Code erhältlich. »Bare Metal« bedeutet, dass die Anwendung ohne Zwischenschicht wie ein Betriebssystem direkt auf den Prozessor zugreift. Die Anwendung ist die einzige Software, die auf dem Mikroprozessor oder Mikrocontroller ausgeführt wird. Egal welche der beiden Laufzeitvarianten der Softwarehersteller einsetzt: Die Programmierschnittstelle bleibt identisch.