改ざん検知やコードの難読化で組込み機器のソフトウェアを保護

IoTの進展によって組込み機器市場が裾野にまで拡がり、ネットワーク接続も進むことで、サイバー攻撃などのリスクが顕在化しています。これらのリスクに対応するためには、改ざん検知やコードの難読化のように、ソフトウェア自体を保護することが対策として挙げられます。当社は、実績のあるソフトウェアプロテクション・ツールを提供してこれらのニーズに応えます。

ソフトウェアへの脅威に対処

社会の至るところで組込み機器が活躍するようになるとともに、それに対するセキュリティリスクも高まってきています。さらに、組込み機器のネットワーク接続が進み、個人が特定されるなどの多様な情報を頻繁に扱うようになったことで、そのリスクは拡大する一方です。

これらのリスクに対応するためには、攻撃の入口となるネットワークへのセキュリティ対策はもちろん必要ですが、機器で動作するソフトウェア自体の保護も考えていかねばなりません。自動車の制御を行う車載ソフトウェアでは、その処理内容を悪意により書き換えられる可能性があり、最悪の場合には、重大な事故につながる可能性があります。また、カード情報を扱う機器のソフトウェア内に暗号鍵やパスワードの情報が記録されていれば、その情報を解析されてカード情報の漏えいが発生し、金銭的な被害にも繋がります。

こうした脅威から組込み機器のソフトウェアを保護するため、当社では、セキュリティ機能を実装する受託開発はもちろん、セキュリティソフトウェアIPやツールの販売、脆弱性管理に関するコンサルティングや教育など、さまざまな取り組みを行っています。

ここでは、これらの取り組みのひとつとして、当社で販売しているインサイドセキュア社のソフトウェアプロテクション・ツールについて紹介します。

改ざんや情報漏えいから機器を守る3つの機能

本ツールは、「Core」と「WhiteBox」という2つの製品から構成され、「実行時の改ざんチェック」、「コードの難読化」、「機密情報の秘匿」の3つの機能を提供します。

(1) 実行時の改ざんチェック(Core)

ソフトウェア実行時に不正な改造が行われていないか、チェックを行う機能です。ソフトウェアを構成しているいくつかの関数にチェック処理を埋め込み、実行中に動的に改ざんの有無を確認します。ソフトウェアの各所に埋め込まれたチェック処理が、各エリアを相互に監視することで全領域の安全性を確実にチェックすることが可能です**(図-1)**。

このようなチェックの方法では、チェック処理の埋め込み量が多過ぎると処理速度の低下に、少な過ぎるとチェック漏れにつながります。つまり、チェック処理の量(安全性)とパフォーマンスはトレードオフの関係にあります。本製品以外にも実行時の改ざんチェック機能を持ったツールは存在しますが、チェック処理の埋め込み箇所やその量は、開発者の経験によって決める必要があります。しかし、本ツールは、ソフトウェアの実行時のデータを取得して自動分析する機能を備えているため、パフォーマンスを低下させずに十分な安全性を確保する最適な配置を自動で行ってくれます。

(2) コードの難読化(Core)

プログラムの流れを複雑にし、攻撃者による不正なソフトウェアの解析を困難にする機能です。例えば、ソフトウェアの中で条件分岐する選択肢の数や、ループの多さなどは、攻撃者がプログラムのバイナリを解析する際の大きな手がかりとなります。コードの難読化機能では、条件式の複雑化や、ループなどの制御構造の変更、ダミーコードの挿入などを組み合わせることで、プログラムの複雑度を増加させます。

難読化の度合いは細かくチューニングでき、性能要件への適合、コードサイズの制限に対応します。処理速度を極力落とさずリバー

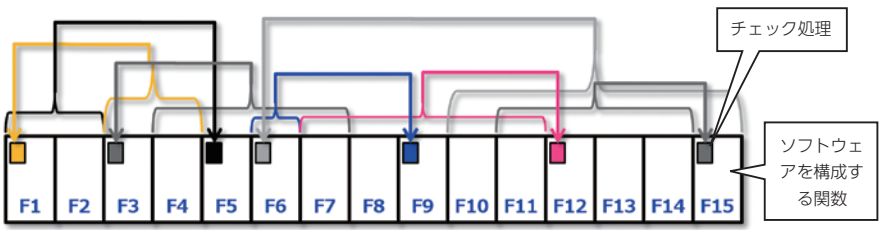

図-1　改ざんチェックの仕組み（複数のチェック処理で相互監視）

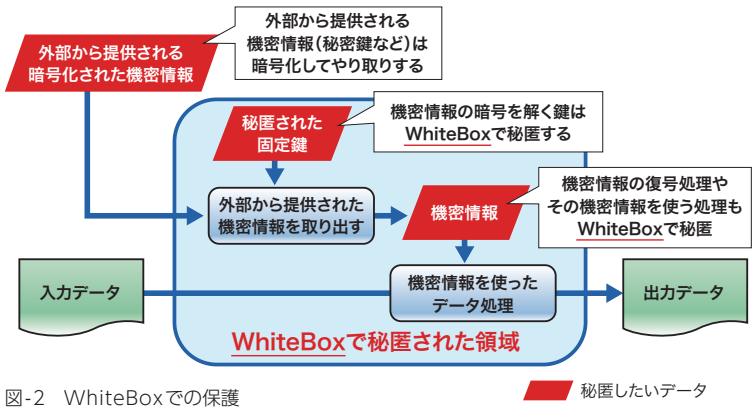

図-2　WhiteBoxでの保護

スエンジニアリングを阻止したい場合に最適な機能と言えます。

(3) 機密情報の秘匿(WhiteBox)

ソフトウェアに埋め込まれた機密情報を強力に秘匿する機能です。例えば、どんなに強力な暗号アルゴリズムを採用しても、その暗号処理を行っているソフトウェアを動的に解析されれば簡単に暗号鍵を取り出されてしまいます。しかし、この秘匿機能を使用すれば、もし攻撃者が動的に解析を行ってソフトウェアの中身を見たとしても、暗号鍵などの機密情報は確実に秘匿され、まったく別のソースコードに変換してくれます。

図-2のように、外部とやり取りする機密情報は暗号化し、それを解く鍵や復号処理をこの機能で秘匿することで、動的解析をされたとしても安全なソフトウェアを実装することが可能です。処理速度の低下やサイズの増加を伴っても、機密情報を守らねばならない場合には最適な機能です。この秘匿処理は関数ごとに適用できるため、絶対に秘匿したい部分に絞って使うことで、影響を最小限に留めることが可能です。

開発環境と統合し2つのフェーズで保護

本ツールは、開発環境の中に統合されます。ツールにより自動的に解析コードと保護コードが挿入され、ソースコードの修正は特別なカスタマイズが必要な場合を除き不要です。

まず第1フェーズである「分析ビルド」では、ソフトウェアの実行時のデータを収集するための機能を埋め込んだ形で一度ソフトウェアを生成します。コンパイルしてソフトウェアが出来上がってから実行し、その動作情報をデータベース化していきます。その後、第2フェーズの「保護ビルド」でデータベース内の動作情報を使って最適な場所にチェック処理を埋め込むことでソフトウェアが完成するイメージです**(図-3)**。

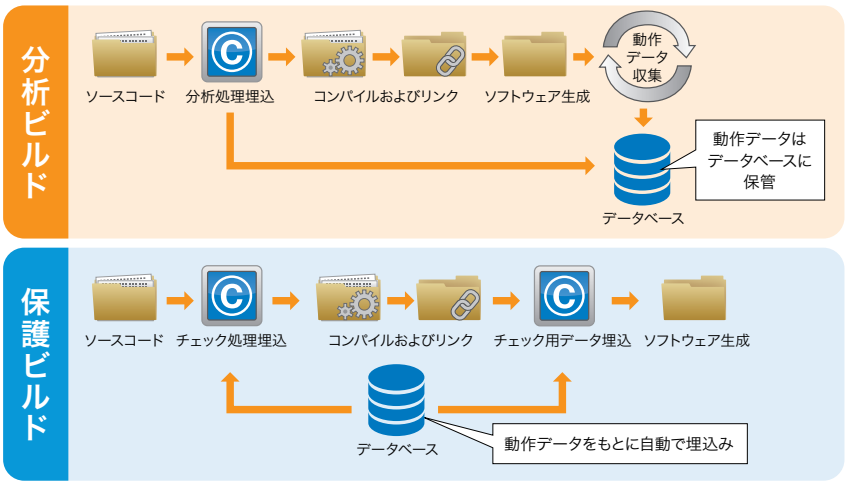

図-3　2つのフェーズで保護

これまで説明した機能を使用して、ソフトウェアを強力に保護しようというのが、当社が提供するソフトウェアプロテクション・ツールです。「鍵の管理は専用のチップ(TPM)を導入しているのでWhiteBoxの機能は必要ない」というお客様や、「ソフトウェアの処理は特殊なことはしていないため難読化は必要ない」といったお客様は、改ざん検知機能のみでも利用可能です。ソフトウェアのアップデート用のアクセス先のURLを書き換えられてしまう、TLS通信用の認証局の証明書に不正なものを追加されてしまう、データを外部に不正に送信する機能が追加されてしまう、といったリスクを防ぐことができます。

IoT時代が到来し、各所にセンサーが配備されるようになってきました。公の目に触れるところで使われながらもセキュリティ監視されていないセンサーや機器が脅威にさらされる危険性が高まり、これらで動作するソフトウェアをいかに保護するかは、今後ますます重要な課題となってくると思われます。

Core/WhiteBoxは、すでにスマートフォンのカード決済アプリケーションをはじめ、モバイル機器やオフィス機器などを扱う情報機器メーカーを中心に豊富な導入実績があります。IoTの進展に伴い、センサー機器にも活用されていくと期待しています。

当社は、本ツールの販売、アプリケーション開発のお手伝いはもちろんのこと、ツールのコンサルティングから教育までを含めたトータルなサポートを提供していきながら、車載などの新しい分野にも拡販していきたいと考えています。

(エンベデッドシステム事業部　関根 正騎)