ホワイトペーパー

The Sentinel Envelope

Table of Contents

はじめに

「ソフトウェアを正当に購入して、利用しようとする顧客に不便を強いることなく、ソフトウェアの不正利用をどのように防ぐかということは、今日のIT業界でソフトウェアベンダーが直面している最大の課題の1つです。」

Copyright infringement of software, 今はインターネット上にリバースエンジニアリングに関する情報が豊富に存在し、すべての人がツールや知識を簡単に入手できる時代です。そのため、ソフトウェアの著作権侵害や不正コピーは以前にも増して容易になっています。ほとんどの国にはソフトウェアに適用される著作権法があります。しかし施行や遵守の度合いは国によってさまざまであり、著作権侵害が多発している国があることは周知の事実です。

ソフトウェアの著作権侵害は増加の一途をたどっています。これは、広く蔓延しており、追跡が難しく、さらには防止や禁止が困難であるためです。ソフトウェアの著作権侵害は収益低下を招くだけでなく、正規に料金を支払う顧客が不正利用による損害を負担させられるという大変な悪影響を及ぼします。自社のソフトウェア製品を積極的に保護しているソフトウェアベンダーは正しい方向に向かっていると言えます。ただし、アプリケーションのセキュリティを損なうおそれのあるハッキング行為は増大の一途をたどっており、これらから十分に保護されていない可能性があります。よくある誤解は、あるアプリケーションが保護された上で配布されていれば、ソフトウェアの著作権侵害や知的財産の盗難から「完全に保護」されるというものです。ここで極めて重要となるアクションは、ソフトウェアベンダーがライセンシングベンダーやハードウェア保護製品のメーカーと連携して、常に最新のセキュリティ対策を導入するとともに、セキュリティレベルを継続的に高めることです。製品ライフサイクルの一環として革新的な保護策やセキュリティ対策の仕組みを導入することで、潜在的な脅威の何歩も先を行くことができます。

ソフトウェアの著作権侵害は増加の一途をたどっています。これは、広く蔓延しており、追跡が難しく、さらには防止や禁止が困難であるためです。ソフトウェアの著作権侵害は収益低下を招くだけでなく、正規に料金を支払う顧客が不正利用による損害を負担させられるという大変な悪影響を及ぼします。

このホワイトペーパーでは、Sentinel Envelopeが提供する、ソフトウェアの著作権侵害や知的財産の盗難からアプリケーションを保護するためのさまざまなメカニズムについて詳しく説明します。

概要

バッカー– 定義と使用法

バッカーは、その名の通り、実行可能ファイルに変更を加え、圧縮のため、またはリバースエンジニアリングからの保護手段として、同等のファイルを新しく作成するツールです。ラッパーの仕組みは、人形を何重にも入れ子にするロシアのマトリョーシカを思い浮かべるとわかりやすいです。ハッキング行うことで、1つ（または複数）のセクションを元の実行可能ファイルに追加するとともに、通常の実行の前にプログラムの「ラッピングを解除」するためのローダーを付加できます。ローダーはOSの一部であり、アプリケーションを読み込むと同時に、アンチデバッギング、トレースの検出、ライセンス管理、バックグラウンドでのチェックといったリアルタイムタスクを実行します。

Sentinel Envelopeは、暗号化とネイティブコードの難読化の組み合わせによって、これまでで最も強力な保護を実現し、貴重な知的財産を確実に保護します。

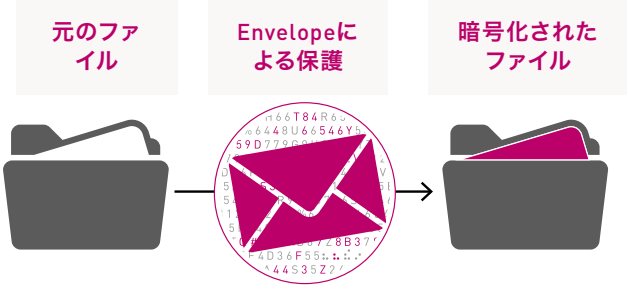
Sentinel Envelope - ワンクリックで簡単に利用できるソリューション

Sentinel Envelopeは、ファイルの暗号化、コードの難読化、システムレベルのアンチデバッギングを通じてソフトウェアのリバースエンジニアリングを防ぐ、自動ファイルバッカーです。実行可能ファイルやダイナミックリンクライブラリのラッピングにより、アルゴリズム、機密情報、専門的なノウハウといった自社の知的財産をクラッカーから保護できます。

Sentinel Envelopeは、ハードウェアベースまたはソフトウェアベースのプロテクションキーにアプリケーションをバインドする保護手段を追加することで、アプリケーションを強固に保護します。

その保護にかかる時間は、デフォルトの保護スキームが選択される場合、わずか数秒です。利用可能なオプションの一部またはすべてを利用することで手順や対策が増える場合、このプロセスは若干長くなりますが、アプリケーションのソースコードにアクセスできないソフトウェアベンダーにとって極めて強力なプラットフォームが実現します。たとえば、未保護のソフトウェアを販売するリセラーやディーラーは、Envelopeのデフォルトの基本設定によって、現地市場向けの製品を簡単かつ迅速に保護したうえで販売できます。

保護されたアプリケーションが起動されると、Sentinel Envelopeのローダー（Envelopeランタイムの一部）がプロテクションキーにクエリを送信し、その存在を確認します。有効なSentinelプロテクションキーが存在する場合、Envelopeのローダーがプロテクションキーの暗号化エンジンと連携して、開発者によってそれまで暗号化されていたアプリケーションファイルを復号化します。Sentinelプロテクションキーが存在しないか無効である場合は、アプリケーションは停止し、実行されません。また、プロテクションキーが利用できない場合は、バイナリが復号化されません。



データファイル暗号化ユーティリティ

アプリケーションの実行可能ファイルの保護とは別に、アプリケーションがアクセスするデータファイルを暗号化することで、知的財産を確実に保護できます。これにより、ハッカーからソフトウェアの知的財産を保護するためのセキュリティ対策がさらに向上します。データファイル暗号化ユーティリティであるDataHASPとSentinel Envelopeが連携して、データファイルを事前に暗号化します。その後、このデータファイルは保護されたアプリケーションによって暗号化または復号化されます。暗号化が完了した後にデータファイルにアクセスできるのは、適切なプロテクションキーが検出された場合のみになります。

機能

Sentinel Envelopeは、ファイルの暗号化、コード難読化、システムレベルのアンチデバッギング、ホワイトボックス暗号化、AppOnChipといった非常に高度な機能によって、リバースエンジニアリングに対するIPの強固な保護を実現します。これらの追加機能を実装することで、ハッカーによる侵害行為が極めて複雑かつ時間のかかるものになります。そのため、ソフトウェアコードがエンドユーザの手に渡るまでの間にコードを漏えいから保護できます。各機能では、ハッカーがアプリケーションの侵害を試みる際に利用するさまざまな手法が考慮されています。

アンチデバッギングとアンチトレーシング手法

デバッガは通常、アプリケーション開発プロセスでバグの検出や問題のトレースを行うためにソフトウェア開発者によって使用されます。しかし、ソフトウェアを不正に利用しようとするハッカーは、その同じデバッガを利用して実装済みの保護コードを検出とトレースし、最終的にはコードの改変、無効化、全削除を目論みます。

Sentinel Envelopeが備える、極めて強力な機能の1つが、実行中のデバッガを常時探索するデバッガ検出機能です。Envelopeは誤った処理を行わせるようなコマンドや偽の情報をデバッガに送信することで、ハッカーに重要な作業を行わせないようにします。さらに、Envelopeではデバッガを簡単に識別できるため、正当なデバッガと不正なデバッガを見分けることができます。また、Sentinel Envelopeはアンチトレーシングツールが起動されたかどうかを検出し、必要に応じて保護されたアプリケーションの実行を停止するように設計されています。

ハッカーと開発者の両方が同じデバッグツールを使用していることは前提に考える必要があります。Sentinel Envelopeでは悪意のない開発者によるデバッグ作業と、危害を加えようとするハッカーによるデバッグ作業を明確に区別できる方法を備えています。それは、デバッガが検出された旨のメッセージを表示し、保護されたアプリケーションの読み込みを防ぐ機能です。開発者はこの段階でデバッガを無効にし、アプリケーションを適切に読み込んで実行できるようにします。一方で、アプリケーションの読み込みと実行の後にデバッガが有効化された場合は、明らかにソフトウェアのクラッキングを試みる“海賊”ソフトウェアの活動であるため、アプリケーションは停止します。

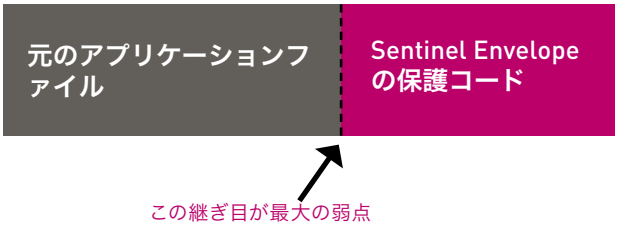
最大の弱点の保護

ラッピングメカニズムによって保護されたアプリケーションの最大の弱点は、アプリケーションファイルと追加された保護コードの間の継ぎ目です。この継ぎ目が破られると、ライセンスが格納されているプロテクションキーへのリンクが切断され、アプリケーションがまったく保護されない状態になってしまいます。そのため、この弱点にはほとんどの攻撃者が侵害を試みます。クラッカーは保護されたファイルを調べ、保護コードと、そのコードと付加されたプロテクションキーがどのようにリンクされているかを解析します。クラッカーがコードを解読しその場所を認識したら、以下のいずれかの方法でクラッキングを行うおそれがあります。

Sentinel Envelopeの機能と利点

- **自動ファイルラッパー** - ファイルの暗号化やネイティブコードの難読化を通じて、ソフトウェアのリバースエンジニアリングに対する強固な保護を実現します。
- **アプリケーションとハードウェアのバインド** - プロテクションキーによって、アプリケーションはハードウェアと密接に結びつけられます。
- **安全な通信チャネル** - Sentinelは、保護されたアプリケーションとプロテクションキーの間で安全な通信チャネルを確保することで、中間者攻撃を防ぎます。Java Envelopeはこの機能によって、ハッカーが通信を傍受してプロテクションキーから送り返されるデータにアクセスすることを防ぎます。
- **実行時の復号化** - Sentinelではすべての.classファイルを仮想マシンに一度に読み込むのではなく、実行時に要求されたときにファイルを復号化するメカニズムになっています。これにより、ハッカーがアプリケーション全体を再構築するような攻撃から守ることができます。
- **各種OSのサポート** - Sentinel Envelopeは、リバースエンジニアリングや改ざんからの知的財産の保護を、各種プラットフォームで実現します。具体的には、Windows（x86）、ARMアーキテクチャのLinux（x86、x86_64）、Android（Javaアプリが対象）です。

- **アプリケーション固有のクラッキング** – 特定のアプリケーションファイルの保護リンクを破ります。
- 一般的なクラッキング – まったく同じ手法がすべてのファイルで繰り返し利用されている場合に、同じメカニズムで保護されているその他すべてのファイルの保護リンクを破ります。したがって、保護されたファイルと追加された保護コードの間の継ぎ目を不明瞭かつトレース不能にし、保護メカニズムを理解しようとするすべての者にとって、大変時間のかかる面倒なプロセスを用意する。防御ではこのことが不可欠です。Sentinel Envelopeが備える最も強力な機能の1つが、この継ぎ目を保護し、保護リンクが破られないようにするための多数の障害を設ける機能です。これは、保護プロセスで複数のレイヤーにわたる保護コードをアプリケーションに追加することで実現されます。こうしたレイヤーは、列車の車両のごとく順番に連結するように特別に設計されたコードです。Sentinel Envelopeは各保護セッションにおいて、元のアプリケーションファイルに追加する際に、コード全体を構成するさまざまなレイヤーが異なる順序で配置されるようにします。

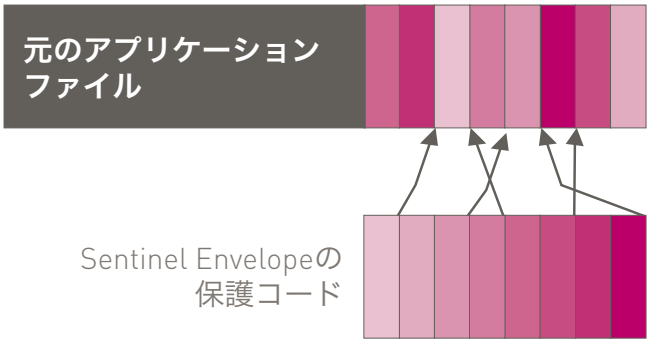


レイヤーの動的配置の方法はEnvelopeのすべての保護セッションで異なるため、保護されたどのファイルも固有のものになります。元のファイルがまったく同じだとしても、保護されたファイルの間で類似点はありません。また、Envelopeのコードにおける最後の命令からアプリケーションコードにおける最初の命令への移行方法は、保護されたアプリケーションの間で異なります。アプリケーションごとに元のコードの開始位置が異なるため、Envelopeで保護されたアプリケーションの継ぎ目はほぼトレース不能になります。保護されたファイル内のさまざまなレイヤーやそのレイアウトを把握できたとしても、別のEnvelopeセッションで保護された同じファイルのレイアウトについては何もわかりません。Envelopeでは侵害行為をさらに困難にするために、レイヤーの配置を変化させるだけでな

く、保護対象のファイルごとにレイヤー数も変化させます。さらに、レイヤーはそれぞれが別の方法で暗号化されます。 そしてアプリケーションの実行時には、各レイヤーがランダムな暗号化キーを使用して、シーケンス内の次のレイヤーを復号化します。

> Sentinel Envelopeではアプリケーションの一部またはアプリケーションファイル全体の暗号化を可能にすることで、個々のニーズに応じたさまざまなセキュリティ構成を実現します。

もしかすると、機能やメカニズムの多さに混乱してきたかもしれませんが、これだけではありません。各レイヤーのコードは、有効な命令の間に挿入されるダミーのオペコードによって難読化されます。これにより、コードについての調査が極めて困難になり、逆アセンブラを使用した保護メカニズムの解析やコードの逆アセンブルは役に立たなくなります。



Sentinel Envelopeはソースコードのラッピングを行うことで、リバースエンジニアリングに対する強固な保護を実現し、アルゴリズムや機密情報といった貴重な知的財産を保護します。Envelopeによって保護されるファイルは、それぞれが異なるランダムシードを使用して暗号化されます。そのため、保護フェーズの後は、元のファイルが同一だったとしてもファイルは大きく異なります。アプリケーションファイルは、複数のブロックに分割されます。こうしたブロックはサイズ変更が可能であり、保護フェーズで開発者があらかじめ決めることができます。各ブロックは、異なる無作為のシードを使用してAESによって暗号化されます。

クラッキング検出時の「挙動の変化」

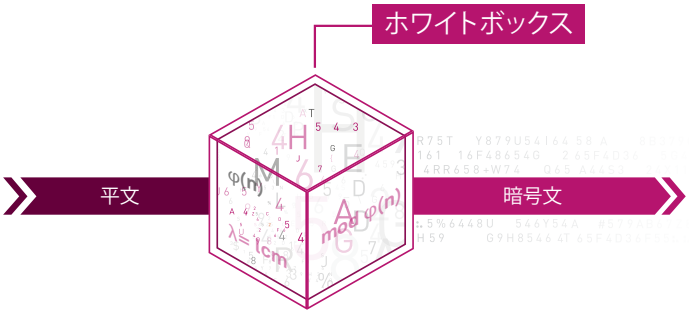
Sentinel Envelopeがデバッガ対策として採用しているもう1つの手法は、「挙動の変化」と呼ばれています。ライセンスが含まれるプロテクションキーでは、OSとデバッガではアプリケーションの実行方法が異なるという点を利用した高度なコード設計が採用されています。（整合性チェックなどで）クラッキングが検出された場合、ソフトウェアの反応が遅くなるため、「原因」と「結果」の間の論理的な関係が絶たれます。 反応が遅くなるとクラッカーは混乱します。クラッキング行為とその行為に対するソフトウェアの否定的な反応の間にある、真の論理的な関係が不明確になるためです。クラッキングが検出された場合にプログラムの機能を低下させるといった挙動は、非常に効果的です。

プライベートAPI

ソフトウェア保護ソリューションを提供する大部分のベンダーは、すべての顧客に同じAPIライブラリを提供しています。そのため、セキュリティ侵害が発生した場合にそのライブラリが単一障害点となります。ジェムアルトでは、はるかに安全なソリューションとしてソフトウェアベンダー（ISV）固有のAPIライブラリを採用しています。こうしたプライベートAPIはジェムアルトのサーバ上でビルドされ、カスタマイズされるため、クラッカーによる攻撃の手は及びません。このAPIなら、各ベンダーは構造的に異なるコンポーネントをアプリケーションに組み込むことができます。このプロセスの一環として、ISVごとに異なる形でカスタマイズされるプライベートAPIは、独自のホワイトボックス暗号化によって強化され、最終的に強力な難読化／保護手法が適用されます。その結果得られるライブラリは、一般的なクラッキングの影響をほとんど受けません。一般にクラッカーが、あるベンダーのAPIライブラリのセキュリティを侵害したとしても、その方法を他のベンダーに対して活かすことは不可能です。Sentinel Envelopeは、開発者のマシンにダウンロードされたAPIのコピーからこうしたプライベートAPIを取得し、保護の際にアプリケーションに組み込みます。 このように強力に保護されたISV固有のAPIは、さらにEnvelopeランタイムによって使用され、ISVの保護されたアプリケーションへの正規のアクセスのみが許可されます。

Sentinelプロテクションキーの定期的な呼び出し

Sentinel Envelopeの大きなメリットは、コンパイル済みのファイルに適用されるため、アプリケーションのソースコードを改変する必要がまったくないという点です。プロテクションキーへの呼び出しは、アプリケーションファイルに追加される保護コード（Envelopeランタイム）によって定期的に実行されます。Sentinelプロテクションキーの存在は暗号化手法によって確認されますが、ISVのセキュリティインテグレータは、Sentinelプロテクションキーをチェックする間隔を設定できます。これは、保護フェーズでISVが全面的に設定できる多くのパラメータの1つにすぎません。

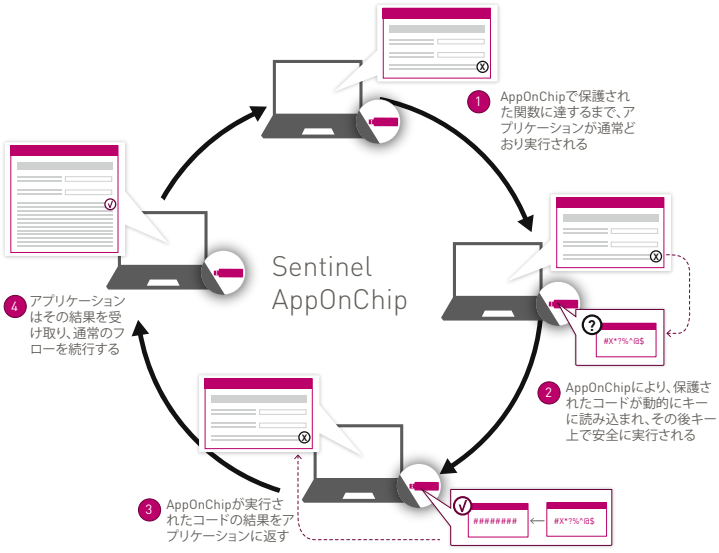


ホワイトボックス暗号化

ジェムアルトはホワイトボックス暗号化の活用において業界のパイオニアであり、保護されたアプリケーションとプロテクションキーの間の通信を攻撃者が解釈するのを防ぐ手段として、通信チャネルを完全に暗号化します。ホワイトボックス暗号化に基づいた安全なチャネルによる通信では、ベンダー固有のコンポーネントを活用し、保護されたバイナリから安全なチャネルの暗号化キーを動的にも静的にも抽出できないようにします。

AppOnChip

Sentinel Envelopeの最もセキュリティを高める機能の1つであるAppOnChipは、Sentinelハードウェアキーとアプリケーションを分離できない形でバインドでき、ソフトウェアベンダーに最も安全なソフトウェア保護ソリューションを提供します。この完全に自動化されたプロセスにより、AppOnChip機能で使用できるコードブロックが含まれる、アプリケーションの一連の関数が提示されます。そして暗号化と署名によって保護されたコードブロックは、ハードウェアキー自体で読み込まれ、実行されます。このような徹底したセキュリティ対策により、Sentinel Envelopeは、市場で最も安全なソフトウェアライセンシングの実装となっています。AppOnChipの機能と利点には、より強力なセキュリティ、容易な導入、ライセンシングの最大限の柔軟性、エンドユーザーに対する透明性、負担のない運用などがあります。



オリジナルエントリポイント（OEP）の保護

オリジナルエントリポイント（OEP）とは、OSがアプリケーションの実行を開始する、アプリケーションの開始アドレスを指します。保護されたアプリケーションのラッピングをクラッカーが解除するには、このアドレスを特定し、ラッパーのコードを削除して、アプリケーションのオリジナルエントリポイントからアプリケーションを起動する必要があります。

> Sentinel Envelopeは、ソリューションを一から開発するための時間や手間をかけることなく、追加設定なしで最も高いレベルのセキュリティを実現します。同時に、エンジニアリングチームがコアな得意分野に専念できるようにします。

Sentinel Envelopeは他のラッパーとは異なり、オリジナルエントリポイントの命令をデフォルトの位置から移動し、Envelopeランタイムコード内で分散配置します。クラッカーが、分散配置されたチャンクからオリジナルエントリポイントを見つけて再構築しようとしても頓挫するはずです。これは、位置やチャンクサイズの不規則性を考えると事実上不可能な操作だからです。

メソッドレベルの保護

Sentinel Envelopeでは、メソッドレベルの保護を定義することで.NETまたはJava実行可能ファイルの保護を強化します。保護対象として.NETアセンブリまたはJavaアセンブリが選択されると、Sentinel Envelopeは個別の保護が可能なメソッドを自動的に決定します。これによって知的財産のセキュリティを確保し、すべてのアプリケーションにおける最適な保護を実現します。.

インポートアドレステーブルの削除

クラッキングを回避するためのさらなる手段として、保護されたアプリケーションで使用する外部DLLの関数のアドレスが格納されている、インポートアドレステーブルを削除するプロセスがあります。元のアプリケーションのラッピングを行うプロセスでインポートアドレステーブルが削除されるため、この情報はディスク上にもメモリ内にも存在しなくなり、Envelopeコード内で分散配置されます。つまり、アドレスのインポート操作はそれぞれ、Envelopeによって保護され、内部的に処理されます。さらに、各インポート操作は異なる難読化コードによって別々のメモリアドレスに分散されます。そのため、クラッカーはパズルのピースを集めるように各インポート操作を個別に解析し、把握しなければなりません。従来のラッパーの場合、クラッカーはインポートアドレステーブルを参照すれば、テーブル内の各エントリの解析が完了したことがわかります。Sentinel Envelopeの場合は、インポートアドレステーブルが使用されず、クラッカーには全体を把握するためのテーブルがありません。そうなると、クラッカーは常に、作業を完了できたかどうか確信を持てなくなります。さらに、Sentinel Envelopeはさまざまな手法によってインポート操作を隠蔽します。これにより、クラッキングされたアプリケーションが後の段階で機能しなくなり、「クラッキングに成功した」かに見えるアプリケーションが結局のところ、まったく使えなくなります。

元のコードとEnvelopeコードの強固なバインド

一般的なラッパーでは、元のアプリケーションコードとラッパーのコードのバインドは行われません。Sentinel Envelopeは、ラッパーと保護されたアプリケーションの仮想的な結び付きを強化します。つまり、保護フェーズにおけるコードフローの解析に基づいて、自身をアプリケーションフローに組み込むのです。これにより、見分けがつかない形で保護手段をアプリケーションに組み込むことができ、攻撃者による削除を防げるようになります。実行時には、制御フローが指定されたアドレスに到達すると、元のアプリケーションのコードが継続的に実行されている間、明示的な実行シーケンスがさまざまな検証・確認処理を実行します。フローに問題がなければ、アプリケーションが実行されます。アプリケーションの整合性に問題がある場合は、プロセスが停止します。

Stolen Bytes

メモリのスナップショットやダンプはよく利用される手法ですが、場合によっては、元のアプリケーションのソースコードに関する情報をクラッカーに与えてしまうおそれがあります。これらを悪用することこそが、保護されたアプリケーションのクラッキングを試みるハッカーにとって重要な第一歩であり、ハッキング対策ソリューションにしてみれば必ず卓越していなければならない領域なのです。

「Stolen Bytes」のコンセプトは、保護されたアプリケーションとEnvelopeコードの間の依存関係を強化することです。これによりクラッカーのデバッグ行為を阻止します。元のソースコードのさまざまな位置からランダムバイトのチャンクを選択して、それをEnvelopeコードの内部にランダムに分散配置します。こうしたコードのチャンク（「Stolen Bytes」）は、保護されたアプリケーションの元のコードが実行されている間、ランダムな新しい位置で実行されます。このメカニズムは、元のアプリケーションコードの終了位置とEnvelopeコードの開始位置の継ぎ目を曖昧にすることで、元のアプリケーションコードとEnvelopeコードの間の依存関係を強化します。

コードとシンボルの難読化

難読化とは、意味のある文字列を文字や数字からなるランダムな文字列に変換するプロセスです。ISVはSentinel Envelopeを利用することで、リバースエンジニアリング対策としての難読化を適用できます。デフォルトでは、保護プロセスの一環として、保護対象の.NETアセンブリに含まれるすべてのシンボル名が難読化されます。さらに、ISVは選択したメソッドのコード全体を難読化することも可能です。コードを難読化するとアプリケーションのパフォーマンスが低下する可能性があるため、これはデフォルトでは選択されていません。ISVは保護するメソッドのリストで保護対象として選択されているかどうかにかかわらず、メソッドにコード難読化を適用できます。

結論

Sentinel Envelopeではアプリケーションの一部またはアプリケーションファイル全体の暗号化を可能にすることで、個々のニーズに応じたさまざまなセキュリティ構成を実現します。セキュリティにはある一定のコストが伴うため、完璧なセキュリティを実現することはできません。したがって、アプリケーション自体に求められるセキュリティレベル（保護すべき対象の価値と、潜在的なリスクを無視した場合に想定される損失）を適切に評価することが極めて重要です。

競合他社による機密情報やノウハウの取得を積極的に防ぐことで、ISVは真の意味で産業スパイ活動を防ぎ、競争上の優位性を高めることができます。Sentinel Envelopeは、暗号化とネイティブコードの難読化の組み合わせによって、これまでで最も強力な保護を実現し、貴重な知的財産を確実に保護します。さらに、Sentinel Envelopeはソリューションを一から開発するための時間や手間をかけることなく、追加設定なしで最も高いレベルのセキュリティを実現するとともに、エンジニアリングチームがコアな得意分野に専念できるようにします。

ジェムアルトSentinelのソフトウェア収益化ソリューション

ジェムアルトはセーフネットを傘下に収めることで、オンプレミス、組み込み、クラウドのソフトウェアベンダーに向けたライセンスおよびエンタイトルメント管理ソリューションで業界をリードするプロバイダーとなりました。ジェムアルトのSentinelは、セキュアで、柔軟性と将来性を兼ね備えたソフトウェア収益化ソリューションとして、ソフトウェア業界で最も信頼されているブランドです。

Sentinelソフトウェア収益化ソリューション製品は、構築や使用が容易で、革新的で、機能にフォーカスしています。顧客の使用規模の大小や技術要件、組織構造の如何を問わず、あらゆる組織ごとに固有のライセンスイネーブルメント、施行、管理に関する要件を満たすように設計されています。クライアントはジェムアルトと共にあってこそ、コピープロテクトおよび知的所有権保護から製品カタログ管理および継続的エンドユーザエクスペリエンス向上まで、あらゆる側面のソフトウェア収益化ライフサイクルに対処することができます。

ジェムアルトの世界中の顧客は、新たな要件に適合して新技術を導入し進化する市場条件に対処してきた実績を重ねてきました。この歴史により、顧客がSentinelを選択することは、顧客が、現在、明日そして将来にわたって自社のビジネスアプローチをよりよく進化させる選択をしたことを意味しています。

Sentinel Envelope、Sentinel EMS、Sentinel HLキーが含まれる無料のSentinel LDK評価キットをダウンロードしていただけます。Sentinel Envelopeの機能の詳細を今すぐご確認いただくには、次のWebページにアクセスしてください。

www5.gemalto.com/sentinel-ldk-trial-jp

ディスカッションにご参加ください

- > Facebook
  www.facebook.com/licensinglive
- > LinkedIn
  bit.ly/LinkedInLicensingLive
- > Twitter
  twitter.com/LicensingLive
- > Google+
  plus.google.com/u/2/106533196287944993975/posts
- > Sentinel ビデオクラウド
  sentinelvideos.safenet-inc.com/
- > Blog
  http://www.licensinglive.com/
- > Sentinel カスタマーコミュニティ
  sentinelcustomer.gemalto.com

お問い合わせ先： すべてのオフィスの所在地と連絡先情報につきましては、
www.gemalto.com/japan/software-monetization をご覧ください。
フォローする： www.licensinglive.com

GEMALTO.COM

gemalto
security to be free