



software protection made simple.

Guarding Against Software Piracy, Source Code Theft and Tampering

A Comprehensive Approach to Software Protection

V.i. Laboratories, Inc.

(Updated August 2008)

Table of Contents

Addressing the Threats to Software Applications..... 4
Traditional Software Protection Approaches 5
CodeArmor® Software Protection Solution 5
How CodeArmor Software Protection Works 6
CodeArmor Intelligence Solution 7
How CodeArmor Intelligence Works 9
Synergy with CodeArmor Software Protection and Intelligence Products..... 9
Summary 9
About V.i. Labs..... 9

Addressing the Threats to Software Applications

As software vendors, corporate entities, and government agencies increasingly market and deploy their applications around the globe, securing the sensitive intellectual property (IP) resident in these applications becomes critical. Organizations need to ensure that software programs are executed the way they are intended and keep IP resident within them safe from piracy, theft, and tampering – regardless of where these applications are located. While security solutions exist to protect the way computers process and store sensitive and classified data, they aren't designed to protect the actual operation of application code and algorithms within computing environments. As a result, valuable applications such as financial systems, gaming, communication systems, voting systems, chip and mechanical design software, and medical equipment are not adequately protected. Moreover, the hacker and pirating communities continue to hone their reverse engineering skills and develop tools such as disassemblers, code tracers, rootkits, and software debugging products to determine how a piece of software works. These techniques have made software piracy, tampering, as well as IP theft increasingly serious problems.

In addition, applications developed using managed frameworks like Microsoft .NET significantly raise the risk of piracy, tampering, and code theft threats as well as increasing the complexity of protecting applications at runtime. Unlike native Windows applications, Microsoft .NET applications are only partially compiled into an Intermediate Language (IL) and contain highly descriptive metadata. This fact means that the deployed code itself can be decompiled into a higher representation source code. Because the framework is a standards-based implementation, there are a multitude of freeware tools available that can decompile .NET code. This greatly reduces the skill level required of a hacker or competitor to tamper and steal the code.

The following threat scenarios illustrate the need for a new software protection approach:

Software vendor piracy threat – A large software vendor releases a new high value computer aided engineering product to the market. Within weeks, software pirates reverse engineer the compiled software and either develop a simple patch to bypass the license enforcement mechanism or, worse, develop their own license key generator that can provide valid license keys to enable all features of the software. The cracked application is made available on the Warez portal web sites, BitTorrent networks, or through vendors on the streets of China. Ultimately the software vendor loses license revenue as unknown businesses and enterprises adopt the software across a slew of emerging markets.

Enterprise tampering threat – A financial institution deploys a client-server application to provide services to partners and high worth individuals. A software client is required to be deployed to remote networks. If the remote client application was tampered with it could lead to data loss and liability risk for the financial institution.

Embedded system provider code theft threat – A provider of high value equipment expands its sales presence into countries where intellectual property protection laws are weak and where new rivals exist. Embedded within the provider's equipment are sensitive software programs that drive the equipment and contain valuable IP that differentiates the product. Because the hardware components themselves are Commercial-off-the-Shelf (COTS) a competitor can reverse engineer the application to remove its dependency on the hardware and repackage the application as its own product.

Gaming software provider tampering threat – A large gaming provider sells and deploys slot machine and casino management software globally. The application is developed in Microsoft .NET. Because .NET can be easily decompiled it is a target of tampering threats.

Traditional Software Protection Approaches

Traditional software security technologies can be grouped into three categories; host based, software vulnerability scanning, and copy protection. Host based solutions treat an application as a “black box” and attempt to safeguard the host environment by analyzing network traffic and application transactions, or by detecting malicious behavior on the host operating system. Software vulnerability scanning solutions analyze application source code to discover vulnerabilities that have been introduced during the development process. These tools are valuable for mitigating security vulnerabilities that are introduced during software development, but they do not harden the application before or during deployment and are not designed to protect applications from reverse engineering. There are myriad copy protection technologies that leverage either dongles or USB keys to bind a user license to an application, wrap encryption around the software itself, or apply code obfuscation to protect applications. Dongle based systems provide an increased level of security by protecting security keys on an external device. However, the cracking community now targets its attacks at this interface and has successfully defeated these systems using device emulators. Other protection technologies encrypt the application software to make reverse engineering difficult, but require source code modifications or the application to be decrypted completely in memory at runtime, leaving it vulnerable to an attack. In addition, many of these approaches do not monitor and detect unauthorized access to software during execution. Code obfuscation helps to prevent reverse engineering but must be introduced during the software development process and requires extensive customization and ongoing engineering support.

To better address piracy, theft and tampering threats – without slowing down the software development process and delaying product releases – a simpler and more comprehensive approach to application security is needed. To meet these requirements V.i. Labs has developed patented software protection and threat detection and reporting technology that are easy to implement and significantly increases the security and integrity of software applications. V.i. Labs CodeArmor platform meets requirements for software protection and threat detection and intelligence needs.

CodeArmor® Software Protection Solution

V.i. Labs' CodeArmor combines comprehensive security with a simplified process for protecting native Windows, Microsoft .NET, and mixed mode software applications. It features strong granular encryption at the subroutine level, anti-debugging, anti-tampering, and secure run-time execution monitoring to secure applications. This unique approach enables organizations to quickly and easily secure mission critical or high value applications without impacting product development or requiring additional security infrastructure. Once protected, applications are persistently secured against tampering, misuse and theft, independent of where they're deployed. CodeArmor can immediately detect and isolate attempts to compromise a system, triggering a range of responses to prevent or frustrate intruders.

System Components

The CodeArmor solution offers a non-invasive protection process, strong application protection, and the flexibility to add application and security extensions to meet the particular security requirements of different deployment scenarios. The system is made up of the following components:

Post Processor – An application that integrates into the software build environment to automatically protect compiled application files (e.g., .exe and related DLLs). The application does not require software development expertise to run and provides both graphical user and command line interfaces to configure and automate protection. After the protection settings are configured the post processor automatically analyzes the application binaries, selects and encrypts the sensitive functions or .NET assemblies requiring protection, and then layers secure execution monitoring, anti-debug, and anti-tampering protection capabilities into the binaries themselves. The post processor obfuscates the protection logic in a unique way each time the protection process is run to ensure that protection is unique across all applications protected by CodeArmor.

Secure Execution Monitor – A collection of protection agents that are embedded into the customer application by the post processor. It provides run-time monitoring, tamper detection, preventive responses, and manages the decryption and re-encryption of application subroutines. The monitoring process verifies application integrity, performs environment checks to prevent malicious activity, prevents reverse engineering, and stops debugging tools from accessing the software programs within the application. For Microsoft .NET environments the execution monitoring ensures that the .NET framework itself has not been tampered with and interfaces with the Secure Container technology to protect Microsoft Intermediate Language (MSIL) code when the application is running.

Secure Container Technology – For applications developed using Microsoft .NET, the CodeArmor protection process uses a secure container technology to create a secure virtual environment to protect the decryption of MSIL code. The container technology is a driver level integration that is only required for .NET applications.

Application and Security Extensions – The post processor provides an architecture that allows application and security extensions to be added to a protected application. These extensions support a wide variety of deployment needs including custom threat response actions, node locking, authentication, and hardware integration.

How CodeArmor Software Protection Works

Post Processor Operation:

To protect an application the software provider configures protection using the CodeArmor Post Processor user interface and saves the configuration within a CodeArmor project file. The configuration specifies which executables and DLL files will be protected as well as the level of protection enabled. Using the command line interface, the CodeArmor administrator integrates the post processor operation into the build process. For native Windows applications .PDB or .MAP files can be used along with the matching files for the post processor to configure advanced Just-In-Time decryption capabilities of specific functions. Once the post processor is initiated, it automatically analyzes the binary files and embeds the protection within the Portable Executable (PE) structure of binaries themselves. For Microsoft .NET applications the Secure Container runtime environment is linked into the application installer along with the protected binaries.

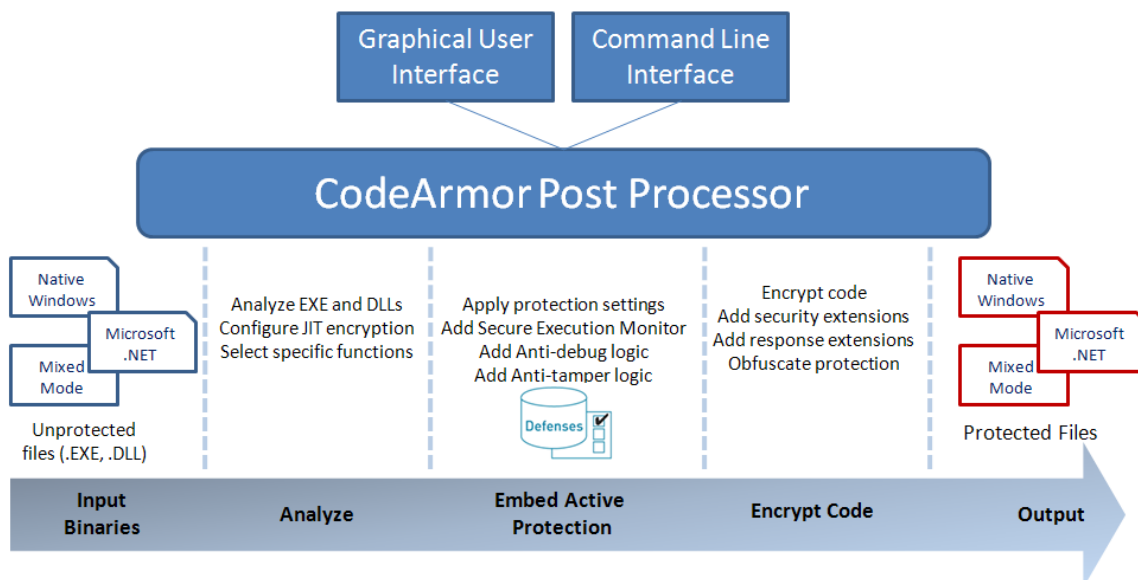


Figure 1: Protecting an Application with the CodeArmor Post Processor

Run-Time Protection Process:

Wherever the application is deployed, the software is encrypted on disk and protected against static tampering. During run-time, the embedded Secure Execution Monitor verifies the application environment for immediate threats, verifies integrity of the configured application files and for native windows applications decrypts the protected functions Just-In-Time (JIT) on a function call basis. The JIT decryption process protects the code in memory by ensuring that only a subset of the application functions are decrypted at any point in time. For .NET applications, CodeArmor creates a virtual Secure Container to protect the decryption of managed code. During the execution of the protected application the execution monitor continuously monitors the environment for threats including kernel and user level debuggers, disassemblers, decompilers, and other tools that attempt to analyze the application. The execution monitor takes interrupt privileges and continually checks for any debug system or code image capture programs that might trace and dump out the decrypted run-time code. If a debug or code capture program is detected, the executable can be halted or use a custom response (i.e. a notification can be sent to the user or audit system).

Within any computation environment, performance degradation is a concern. The CodeArmor runtime process is highly efficient and optimized for speed. The impact to application performance is normally between 2 and 5% CPU utilization when compared to its unprotected original. Performance can be tuned and optimized using the CodeArmor protection level configurations.

CodeArmor Intelligence Solution

CodeArmor Intelligence allows software providers to build threat detection and reporting capability into their applications and to take action using that data. This product is applicable to any organization that needs real time notification and reporting of threats for deployed software. For software vendors, the CodeArmor Intelligence solution offers a turnkey piracy lead generation capability that quantifies the impact of piracy for a specific application and generates and reports forensic data to identify infringing organizations allowing sales and legal action to recover license revenue directly from those organizations. Unlike CodeArmor Software Protection, the Intelligence product does not prevent the “cracking” of the software release and instead remains dormant in the application until the software is actually used and piracy is detected. Once activated, the data is transparently and securely delivered to a gateway and then to the reporting interface for action.

CodeArmor Intelligence System Components

The CodeArmor Intelligence solution is comprised of the following components:

Software Development Kit (SDK) – The SDK provides a flexible programming interface to enable software providers to embed threat detection, data collection, and reporting functions within application software. A utility is used to configure the runtime detection and reporting operation. Configuration options allow the detection and reporting operation to be triggered based on the tampering of specific executables and DLLs or other custom threats. In addition, the reporting function can be delayed and only activated in cases where the software application is actually used. This minimizes the risk of detection of the reporting function and ensures that information is only collected from organizations infringing on the actual software IP. In addition, CodeArmor Intelligence offers multiple levels of reporting including stealth approaches to maximize data reporting. Data collection is designed to identify the organization using the software and can be configured to gather user attributes. The SDK allows software providers to input application specific data into the reporting process as well. Once collected the data is encrypted for secure transmission to the Web gateway.

Web gateway – A Java servlet application acts as a gateway between the CodeArmor Intelligence enabled application and the reporting interface. The servlet can be deployed in the software provider's network using Microsoft IIS or Apache Web servers on Windows and UNIX operating systems. The gateway receives data from the Intelligence instrumented application, decrypts it, extends the information using reverse DNS and WHOIS services, and then logs the data to an XML file. The XML data log allows integration of the data into internal reporting systems. The data is then transformed into infringement records and securely posted via SSL and an authenticated session to the software provider's Salesforce.com instance.

CodeArmor Intelligence Plug-in for Salesforce.com – A plug-in to Salesforce.com enables software providers to organize collected infringement data into a format that can easily be filtered, reported, and managed. The plug-in provides initial workflow and notification rules as well as a piracy lead dashboard. Because the data is available through the Salesforce.com web interface, legal, product management, sales, and other compliance representatives within the software providers organization can easily access and export the data. Alternatively, the Salesforce.com APIs and partner infrastructure allow information to be directly integrated with internal CRM systems (e.g., Seibel, SAP, Oracle).

CodeArmor Intelligence System & Component Overview

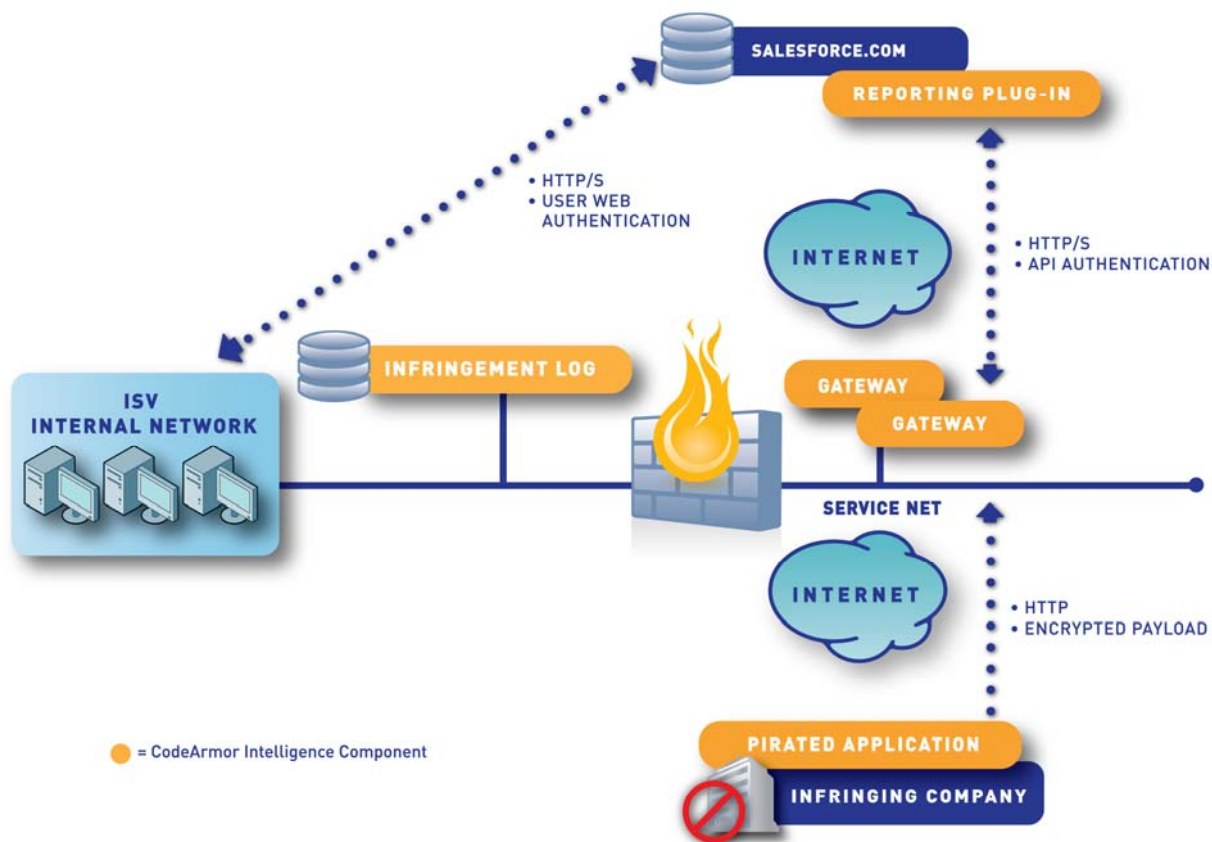


Figure 2: CodeArmor Intelligence System

How CodeArmor Intelligence Works

In an anti-piracy example, the software vendor uses the CodeArmor Intelligence SDK to instrument the software application. Once released, the piracy groups release their “cracked” version with the vendor’s license management application disabled. The cracked release is then distributed through the various software piracy channels (i.e. P2P, Web, Warez). End users obtain the software and install it in their network. When end users operate specific functions (ones that signify actual use of the software), CodeArmor Intelligence is triggered if it detects tampering or other events and then it begins reporting. After successive use (configurable) CodeArmor Intelligence collects machine, network, application, and other customizable data, encrypts it, and then sends it to the Web gateway.

Once the data is sent, the Web gateway adds other data from reverse DNS and WHOIS services to the infringement event, logs it, and then authenticates itself with the vendors Salesforce.com instance and posts the data. The software provider organization is then able to access and manage the data within Salesforce.com via a secure Web browser connection.

Synergy with CodeArmor Software Protection and Intelligence Products

No software protection technology can deliver absolute security. Software providers should focus on extending the time and resources required to tamper or reverse engineer the application. However, by coupling CodeArmor Intelligence with CodeArmor Software Protection, software providers can maximize protection. For example, if the software protection is compromised, the CodeArmor Intelligence capability can still be resident within the software and activate to provide threat notification and intelligence on the use of the compromised application. By configuring CodeArmor Intelligence to detect the threat only when the application is actually used, this will maximize the chance of it not being detected by the individuals attempting to break software protection.

Summary

The CodeArmor platform provides a layer of active protection and threat detection that significantly deters piracy, tampering, and code theft. V.i. Labs’ patented technology automatically and easily secures applications after the development process is complete, eliminating the need for dedicated software development resources or impacting product time-to-market. Expanding on protection, CodeArmor Intelligence offers flexible threat detection and reporting product that can be applied by software vendors and enterprise software providers.

Unlike other security approaches that are designed to secure network elements, encrypt data, detect network intrusions, or provide a static wrapper of encryption around an application, V.i. Lab’s CodeArmor protects the actual operation of application code and algorithms within computing environments. This approach ensures that software applications are executed as intended and, ultimately, guards against financial loss, reputation damage, and threats to mission critical software.

About V.i. Labs

V.i. Labs provides software protection solutions that prevent the misappropriation and theft of intellectual property resident in software applications. Through V.i. Labs’ patented technology, software vendors, embedded system providers, enterprise organizations and government agencies are able to easily detect, gather intelligence, and protect against the threat of piracy, tampering and theft, independent of where the applications are distributed. V.i. Labs is privately held and is headquartered in Waltham, MA. For more information please visit <http://www.vilabs.com>.

To learn more about V.i. Labs’ software protection solutions, please contact us at:

Web: www.vilabs.com | Phone: 781.398.3400 | Email: info@vilabs.com