

ir.deta

**THE CASE
FOR CONNECTED
APPLICATION
-CENTRIC
SOFTWARE
SECURITY**

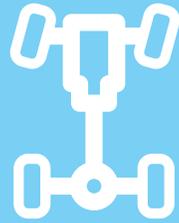
— Clifford Liem —

(Abridged version)



TABLE OF CONTENTS

Chapter 1: Executive summary	3
Chapter 2: Introduction	5
Chapter 3: Traditional software security	7
Chapter 4: Application-centric protection technology for today's connected world	9
Chapter 5: Connecting the dots into a comprehensive solution	13

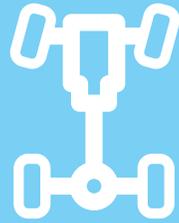


Chapter 1

Executive Summary

Executive summary

For years, the technology industry has been thinking about security from the outside-in. The time has come to be securing applications from the inside-out. A number of modern software protection technologies aid with the protection scheme, including whitebox cryptography, code transformations for anti-reversing and anti-hacking, anchoring to a trusted source, and monitoring/diversity/updates. Whether your application is mobile-oriented, or driven from a web browser on a desktop/laptop, Irdeto provides an extensive solution covering back-end verification services in the cloud, secure encrypted communication, and self-protection of the client.



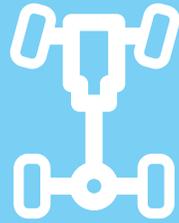
Chapter 2

Introduction

Introduction

Not long ago, entrepreneurs presented a ‘Barracuda Security Device’ on an episode of Dragon’s Den, where ideas are pitched and business cases are shown to potential investors for their products. The pitch demonstrated a burglar trying to steal a computer, but then being thwarted by the audible alarm system (*beep... beep...*). The reactions were swift and blunt, asking ‘How would the device fit in a laptop?’, ‘The data is not even on the PC. It’s on the server.’, ‘Did you come here in a time-warp machine from the 80s?’ It’s easy to laugh at this pitch seeing how out-of-touch the entrepreneurs were, but while it is very entertaining, the story teaches us an important lesson about security. People can easily fall into outdated ways of thinking in their belief of where security problems lie.

Today’s applications run in much different environments than those of 10 to 20 years ago. We have apps running on smartphones, virtual machines, scripts running on browsers, millions of IoT devices all connecting to the cloud, which run a multitude of services. Isn’t there a better way to think about security? What if applications could be self-protected? Why can’t an application carry security with it and be protected from the inside-out?



Chapter 3

Traditional Software Security

Traditional software security

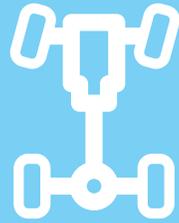
Traditional security practices provide protection to cursory intrusion and simple observation; however, there is a common understanding that a determined attacker will always be able to break the security of most software with enough time and resources at his disposal.

Perimeter security approaches like firewalls concentrate on preventing or detecting threats entering networks of an organization. “Everything on the inside is trusted”. Threats can arise from inside the network from malware, malicious insiders, IoT devices, etc. Additionally, with services in the cloud, more and more critical company data resides off of the company’s internal network.

Signature-based security like virus-scanning looks for known bad data based upon previous identical attacks. This long-standing blacklisting approach is losing the battle against new malware, as in 2014, 317 million pieces of malware were created (nearly 1 million pieces every day). Additionally, today, we have mobile devices, virtual machines, and cloud services. Looking at this problem with the traditional outside-in view may not scale in the future, and more is needed to protect critical systems running in today’s cloud-mobile-web applications.

Platform based security seeks to prevent data leakage, relying on OS native security or security support on the system level, rather than in the application itself. For example, Android uses sandboxing to separate processes with DAC/MAC (Discretionary/Mandatory Access Control) methods. Nevertheless, Android rooting methods still abound. Despite being in a walled-garden, the situation with Apple iOS and OSX has not been much different. Sandboxing techniques and a proprietary Keychain have been the main security techniques, yet these have been shown to be compromised, and jail-breaking methods continue to exist doing little to improve security.

It’s time for additional approaches to protect the application from malicious behaviour on the host.



Chapter 4

Application-Centric Protection Technology for Today's Connected World

Application-centric protection technology for today's connected world

Many applications are striving to be on the primary mobile devices (i.e. iOS, Android) as well as the main desktop environments on web browsers (i.e. Windows, Mac, Linux). This configuration of back-end + web + mobile would cover a large percentage of applications in today's connected world.

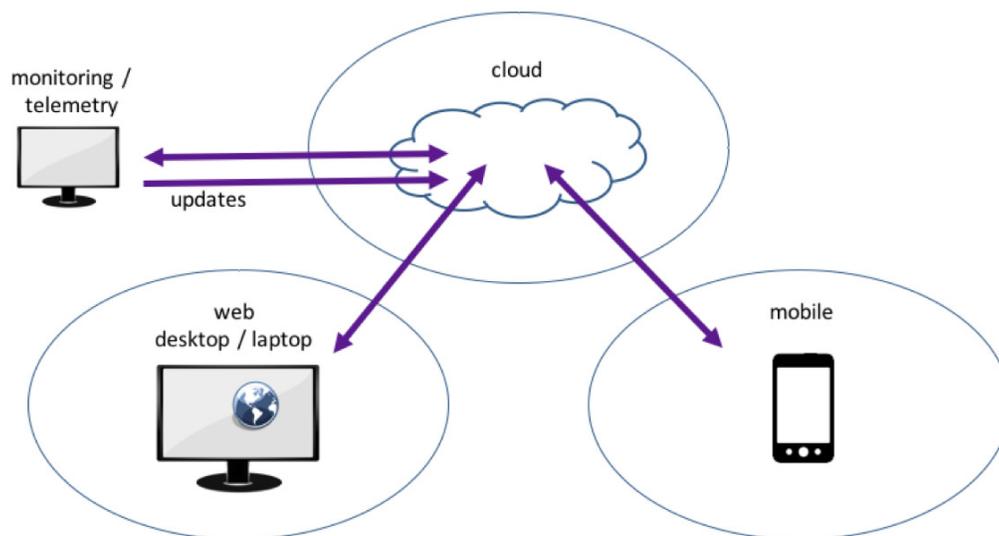


Figure 1 Connected Application Topology

Using an attack tree model as a basis for your application security, the architect and developers need reliable security tools in order to meet the high demands of performance and protection for the essential parts of the system.

Whitebox Cryptography: Hiding keys even while running

Whitebox describes an attack context where the adversary has full visibility and control over the running software; and therefore, presents a very challenging environment for protecting ciphers. While cryptography is commonly used in everyday products, very little attention is being placed on securing the running software in an extremely hostile environment like an open, unprotected environment that helps an attacker. If an intruder can pull a key out of a static

database or simply set a break-point in a running program to reveal a key, then all is lost, especially if that key is used by every instance of the software.

Irdeto has a great history of whitebox cryptography implementations for algorithms like RSA, ECC, AES, 3DES, SHA2, and more, utilizing advanced techniques. Even during running code, keys are never revealed in the execution. The concealment is complete and includes both full keys and round keys, where attacks commonly start. Irdeto provides Whitebox Cryptography implementations in all of its product and service solutions.

Code transformations: Obfuscation, anti-reversing, anti-hacking, and diversity

While cryptography is important, it must be placed inside a large set of logic and program structure which performs the other essential parts of the application. This code is ideally protected from branch-jamming, reverse-engineering, and replaying attacks.

Code obfuscation aims to make a program unintelligible to an attacker; however, the majority of obfuscation tools in the public domain only make cursory changes over the variable naming and the structure of the program, which does very little as a barrier for attacking the running software. On the other hand, transforming a program so that it is difficult to reverse, difficult to debug dynamically, and resistant to modifying and replaying is a much different problem than simple obfuscation. Irdeto provides code transformation for obfuscation, anti-reversing, and anti-hacking with the Cloakware Technology, used in all product and service solutions.

Ensuring integrity and anchoring to a trusted source

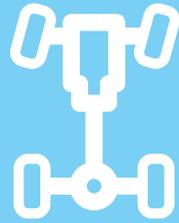
Integrity Verification is the automatic form of making sure that the code is the same as the code that is originally deployed. This tamper resistance technique covers the program by first signing the code segments with a set of digital signatures in either native or web implementation. In a connected scenario, the build-time signatures (i.e. golden signatures) can be kept on the server back-end, where they have a lower risk of compromise. The signatures of the code can be calculated at run-time on the potentially hostile device, where responses can be handled on the back-end in the cloud, which is isolated from an attacker's viewpoint on the client.

Irdeto provides both Native Integrity Verification and Tethered Integrity Verification (Tethered IV) with its product and service solutions.

Secure API and secure connections end to end

Session encryption is common practice in communications to a trusted back-end; however, attacks on certificate chains including forgery may be a concern when relying solely on https connections. Our approach is to use an additional data encryption layer which uses a Fixed-Key WhiteBox AES implementation where the encryption values take on a unique data encoding only known to the trusted back-end. The key is difficult to identify and extract from the implementation; and further, the key and code are unique to a particular client instance.

Unique client communications and diverse instances are implemented with Cloakware Technology (WhiteBox Crypto & Code Transformations) and used in Irdeto's core products and services.



Chapter 5

Connecting the Dots into a Comprehensive Solution

Connecting the dots into a comprehensive solution

Using the software protection technology and techniques described above, we can build a solution with secure communication, tamper resistance, and detection integrated into a cloud service. When combining whitebox cryptography, code transformations, tamper resistance and tethered IV, we are able to build-in security as an integral part of the nature of the code, preventing the security and functionality from being separated. This complements existing methods by going beneath the skin and surgically combining security to the running code. Furthermore, providing diversified client instances combined with telemetry and updates, rounds out the application-centric solution for a highly available, highly protected, and highly maintainable deployment.

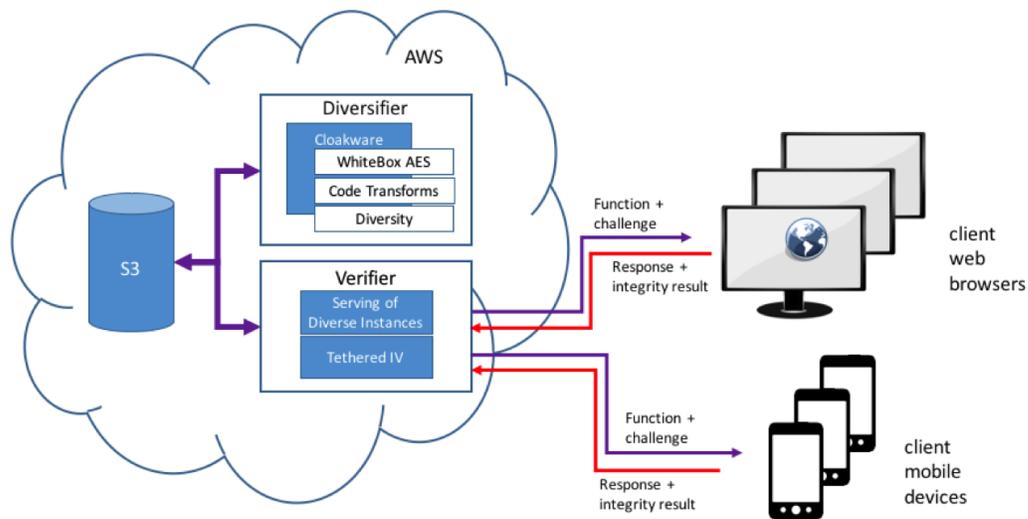


Figure 2 Cloaked.JS/Cloaked.Apps Cloud Services

While there are a multitude of application-types out there, a large percentage have similar topologies with mobile or web clients connecting to back-end cloud services for essential data. Starting with self-protected apps, this connection topology allows clients to be verified through secure communication and tethering to the back-end. The combination of this system with monitoring and updates provides a full security system.

Please contact Irdeto at marketing@irdeto.com for more information on Cloakware Technology and Cloaked.JS/Cloaked.Apps Cloud Services.

References

[IRD1] Irdeto, "The Case for Connected Application-Centric Software Security", full version.