

Sustainable Device Security:

Breaking The Hacker Business Model With Software Security

June 2011

Table of Contents

1. Introduction	3
2. Requirements for Sustainable Security	8
3. Diversity Reduces Impact of a Breach	
4. Renewability	
5. Summary	

1. Introduction

This position paper is for mobile phone and consumer electronics (CE) manufacturers and their reference design suppliers who are enabling devices to play back premium content.

An effective security solution for smartphones should address Integration, Attack Resistance and Attack Mitigation, which together enable sustainable device security and break the hacker business model. Any form of security—hardware or software—can eventually be cracked. Relying solely on tamper-resistant hardware for security may provide strong initial attack resistance but does nothing to reduce the impact of the inevitable successful attack.

Software security solutions are unique in that they enable diversity and renewability that together increase initial attack resistance and minimize the scope and longevity of a successful attack. This combination reduces hacker incentive and minimizes the hacker's impact on the smartphone. Only through the flexibility of software security can manufacturers achieve sustainable device security while quickly bringing new products to market.

THE HACKER THREAT

Device security is increasingly important as digital content becomes pervasive. An increasing amount of personal information, confidential corporate information, trade secrets, and premium paid content must be protected from theft and alteration.

There has been and will continue to be an explosion of devices handling, storing, and transferring valuable digital information. These devices often run standard

operating systems on widely available hardware platforms. The combination of a well-understood operating system and a familiar hardware platform exposes the device to a wider range of attack tools and hackers, dramatically increasing the probability of device security being compromised.

The content industry depends on premium (or paid-for) digital content to be securely distributed, stored and played by the consumer. Hacking and piracy cost the media industry hundreds of millions of dollars each year. The onus is on the device manufacturer to create secure devices that meet anti-piracy requirements. When a device does not adequately protect content, the capability to play back premium content on that device may be revoked.

To combat such attacks and reduce the risk of revocation, device manufacturers are looking for robust technical solutions that protect their revenues, their customers and their brand. This paper describes how device manufacturers can achieve sustainable and cost effective device security.

BREAKING THE HACKER BUSINESS MODEL

Commercial hackers exist because it is financially attractive for them to be in business: the benefits of a successful attack exceed the cost to develop and implement the attack.

If a device manufacturer can make it difficult for hackers to have a sustainable business, the vast majority of hackers will focus their attention elsewhere—for example, on other devices. There are several ways to break the hacker's business model:

- Make the hacking process more difficult and time consuming
- Limit the scope of a successful attack to a subset of the overall installed base
- Limit the longevity of a successful attack

Making it more expensive to successfully compromise a device is a useful and common approach in hardware and software security models. However, making it difficult for hackers to gain significant revenues from an exploit is more effective. Irdeto believes that only solutions that address all three of the points above provide a sustainable competitive advantage in the war against hackers. In the end, it is not about perfect security, but making hacking uneconomical.

Hardware-based security is sometimes harder to break in the first place, but once a successful attack is found, the whole device class is susceptible to the attack. Recovering from such an attack requires a significant amount of time and money. In short, hardware is often subject to a “class attack” and does not address the second and third points.

Irdeto’s software protection technology makes software structurally diverse in an automated fashion. Software diversity ensures that an automated attack (for example, a utility) cannot be applied broadly. There are many ways to deploy diversity: between customers, between devices, between software releases, and between end users. All such approaches reduce the benefit gained by a successful attack. The incentive to attack a system decreases significantly as diversity increases, as shown in Figure 1.

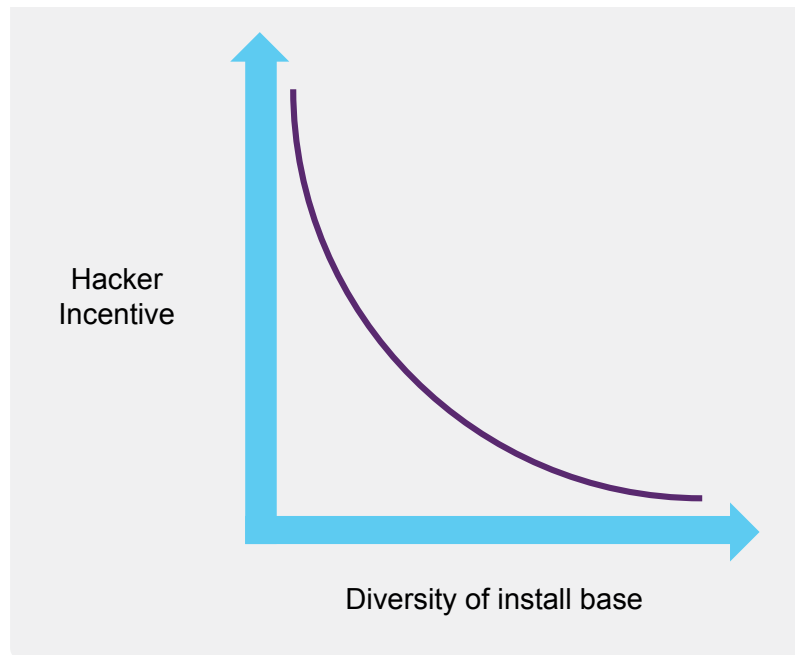


Figure 1. Breaking the hacker business model

Software can also be easily and quickly updated to respond to a successful attack. Updates should be diverse too. Otherwise, the response will be easy to analyze and defeat through a follow-up attack.

Software diversity (in time and in space) and software renewability address the latter two points that hardware does not. They limit the scope and longevity (breadth and duration) of an attack, and thus break the hacker business model, reducing the incentive to hack the device.

The use of diversity prevents class breaks associated with hardware security. Figure 2 and Figure 3 show the differences between a hardware-only security solution and a software security solution which leverages software diversity.

In the hardware-only scenario (Figure 2), once a successful attack is discovered, the attack is quickly packaged as a tool to be distributed to others. The attack has the potential to impact 100% of the installed base in very little time. When a

fix to the attack is eventually available, the remediation process requires a lot of time to deploy and generally does not apply to devices already in the field.

In the software-based security scenario (Figure 3), the manufacturer used diversity to ensure that only 10% of the installed base would have the same software instance (e.g. each device in the field has one of ten diverse software images). When a successful attack is developed, it only affects devices that share the same software image. Also, because software is easily renewable, the time to fix is much shorter and the remediation process happens quickly through the software upgrade/update process.

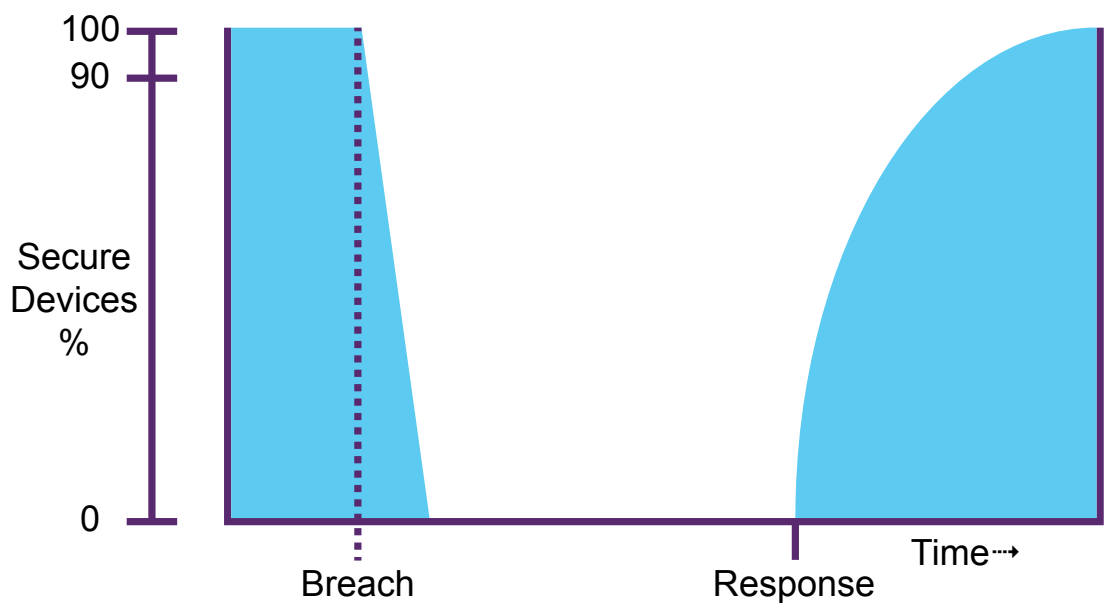


Figure 2. Installed-base security using hardware security only

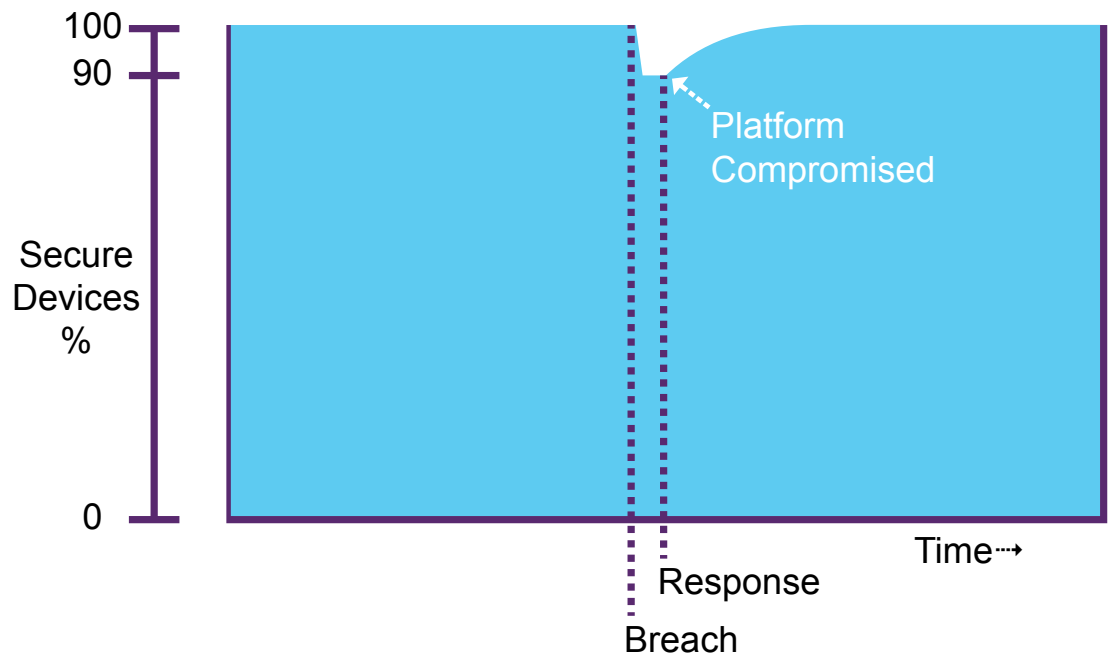


Figure 3. Installed-base security using software security

2. Requirements for sustainable security

Digital media devices consist of both hardware and software components. At some location in the system architecture, the processing hardware interfaces with the operating system and device drivers. These interface points are typically the weakness in the security architecture and they require software-based security mechanisms. As all devices become more complex and full-featured, there is no question that sensitive software will eventually operate on valuable data.

The history of device security suggests that most platforms are eventually compromised, and both hardware and software can be successfully attacked given enough time and resources. For experienced device manufacturers, it is not a question of “if”, but rather “when” the security will be compromised. Thus, to create a sustainable security model, it is useful to separate the security requirements into the following three phases, as shown in Figure 4.

1. Integration: A security solution should be easy to integrate and be compatible across the product portfolio to minimize development costs.
2. Attack Resistance: A security solution should offer strong initial attack resistance and be updateable to offer defense against new-found threats.
3. Attack Mitigation: If a successful attack occurs there should be mechanisms to minimize the impact of the attack and mechanisms to quickly recover from the attack. This is important because an attack may lead to revocation of the devices' DRM licenses, which would harm service revenue and brand perception.

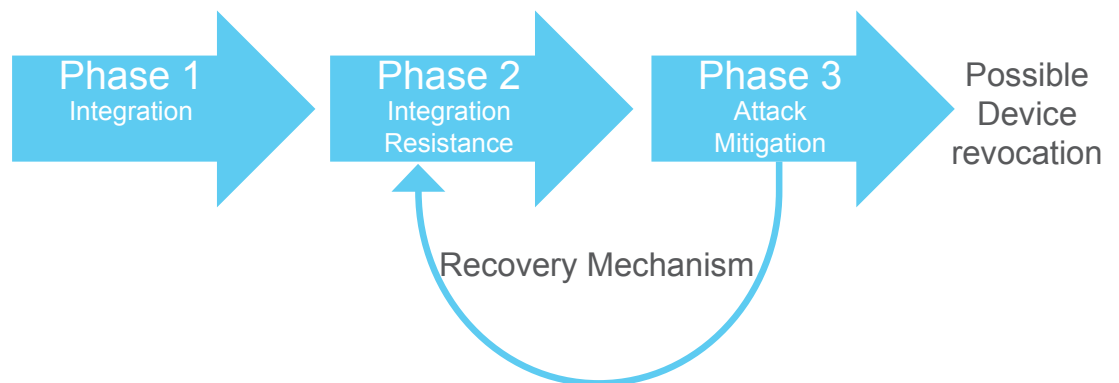


Figure 4. Device security requirements

PHASE 1: INTEGRATION

A security solution should be easy to integrate and be compatible across the product portfolio to minimize development costs.

INTEGRATION SPEED AND EFFORT

To utilize a hardware-based security solution, the security software designers have to become familiar with the APIs and drivers before they can make use of the underlying hardware. The designers also have to develop/modify their software to fit this security framework. This requires significant effort and time and is error-prone. For example, devices based on Symbian 9.1 require application

re-engineering to leverage platform security features, and Texas Instruments' Secure Mode has faced similar implementation challenges in the smart phone market. Software-based solutions can be self-contained and are deployed faster and easier.

In contrast to self-contained secure software, implementing hardware security requires many providers to work together. This ecosystem itself introduces new threats. Test certificates and emulators can be used as hacking tools. Even when the development process is completely trustworthy, the security of the end device depends on the security of your suppliers and fellow licensees as well.

Even with a comprehensive hardware and platform based security solution, it is difficult to implement all of the sensitive code inside the security environment. While some content protection systems are migrating to standard cryptography (based on RSA and AES), many systems will not (for example, DTCP, CPRM, CSS). These non-standard cases require software protection of keys and algorithms and conformance with Robustness Rules. Not only is static protection required, so is protection from dynamic attacks such as buffer and heap overflows.

The end result is that a portion of the solution always relies on software security.

PLATFORM INDEPENDENCE

Consumers want to share their information across disparate devices, whether a PC, a media player, or a mobile phone. Many original equipment manufacturers (OEMs) and independent software vendors (ISVs) already support multiple platforms, each with their own security features and nuances. Software solutions are, for the most part, hardware-independent and can be used consistently across the product portfolio. This level of consistency improves security,

reduces development costs, avoids vendor lock-in, improves time to market, and reduces future support costs while meeting customer demand. Portable software combined with software security is the most practical and scalable solution.

FORWARD/BACKWARD COMPATIBILITY

A security solution must cost-effectively support new devices as well as the existing installed base of devices. Software security solutions can be applied to high-end and low-end devices while providing upgrades to the existing installed base.

Deploying new hardware into the market takes time. The market requirements for security and features often change along the way. Experienced manufacturers recognize the flexibility provided by the combination of software and a general-purpose processor to get high margin innovative devices to market quickly. Not only is software security needed immediately, it is the best answer for future-proofing your new devices.

PHASE 2: ATTACK RESISTANCE

A security solution should offer strong initial attack resistance and be updateable to offer defense against new-found threats.

INITIAL ATTACK RESISTANCE

Software security uses a layered approach to maximize attack resistance. Code transformations can be combined with encryption, integrity verification, and anti-debug techniques to enable a very high level of attack resistance.

While there is the perception that hardware has a higher level of initial attack

resistance than software solutions, it is important to note that hardware by itself does not provide a secure solution. The strength of the security implementation depends on how the hardware is integrated with other hardware and system software. The resulting system has a broad attack surface and the overall level of security is platform-specific.

Hardware also takes much more time to develop and launch into the market. Features are already a few years old by the time a device is delivered to market. The device then must survive three to six years in the field. It is difficult for the original designer to fully anticipate future threats and newly discovered weaknesses as needed to create a secure solution. It is equally difficult to anticipate market requirements. The movers and shakers in the consumer electronics market tend to be companies that leverage software for its flexibility, as needed to navigate the quickly changing landscape.

UPDATEABILITY

Software updates and software renewal is becoming common for network-connected devices. Software is renewed for three reasons:

- Bug fixing
- New feature deployment
- Security enhancements

Over time, attack methods and the state-of-the-art for hacking tools will change. A security solution should be proactively upgradable to counter existing and emerging attacks. Because software solutions can be self-contained, they can be upgraded easily without significantly impacting other system software.

Renewability reduces the longevity or duration of an attack. By limiting the attack's useful lifespan, it also reduces the benefit to be gained. Renewable software security, preferably combined with diversity, is the reason why software will be more secure over time than hardware-based systems.

HYBRID SUPPORT

Software security solutions can work in conjunction with hardware-based security. Hardware can be used to supply keys and other secrets to the software solution which then can expand the chain of trust. The reality is that software requires hardware to execute, and vice versa. Hardware security alone is not a complete security solution because at some point the hardware interfaces to the OS or other software components. Most advanced security systems under development today leverage the physical identification provided by hardware and the responsiveness and low distribution cost of software.

PHASE 3: ATTACK MITIGATION

If a successful attack occurs, there should be mechanisms to minimize the impact of the attack, and mechanisms to quickly respond to the attack.

3. Diversity reduces impact of a breach

Hardware security may provide strong initial resistance to attack. However, when it is broken, the security for the entire system fails. Unlike software, the install base is homogeneous and the same attack works everywhere. Recovery from a successful attack is typically difficult, expensive, and time consuming. Meanwhile, there is the possibility that the device will be revoked, impacting services and profits.

Software diversity mitigates the impact of an attack, or scope, when used both in time (between software releases) as well as in space (at any one time within an install base).

4. Renewability

When there is a successful attack, the fastest response is to update the software via either the next software release, or renewable software. The recovery mechanisms available depend on the type of exploit. When there is a bug in the hardware, the exploit is patched until the next version of hardware becomes available—which may take two years for an ASIC process. Once new hardware is available, it also takes time to be physically deployed into the customer base.

Software can be renewed quickly and inexpensively. If the lessons learned in the area of device and hardware based security (e.g. smart cards) are examined, one can see that software updates are an essential component. Smart OEMs leverage software updates from the beginning.

5. Summary

Other than the possibility of offering stronger initial attack resistance, hardware security adds minimal value through the security life cycle of the product while adding significant cost, as shown in the following table.

History shows that both hardware and software can be hacked, resulting in the end device being compromised. Hardware security aims to provide stronger initial attack resistance but does nothing to reduce the scope or longevity of a

successful attack. As the table shows, software solutions are unique in that they add value at all stages of the security life cycle.

Software flexibility reduces integration effort and maximizes compatibility. Upgradeability and breadth of security techniques provide robust initial attack resistance. Finally, and most importantly, software diversity and response mechanisms together mitigate both the scope and longevity of a successful attack. Under the threat of device revocation by a content protection system, software security is a must for device manufacturers that wish to protect their service revenue and brand.

		SOFTWARE	HARDWARE
PHASE 1 INTEGRATION	Ease and speed of intergration	●	
	Platform independance	●	
	Forward/backward compatibility	●	
PHASE 2 ATTACK RESISTANCE	Initial attack resistance	<i>Depends on Implementation</i>	
	Updateability	●	●
	Hybrid support	●	
PHASE 3 ATTACK MITIGATION	Deversity	●	
	Renewability	●	

Figure 5. Sustainable security

REFERENCES

“Diversity via Code Transformations: A Solution for NGNA Renewable Security”, Yongxin Zhou and Alec Main, In 2006 NCTA Technical Papers™

“Software Piracy Prevention through Diversity”. Bertrand Anckaert, Bjorn De Sutter, and Koen De Bosschere. In Proceedings of the 4th ACM Workshop on Digital Rights Management, 2004.

“Application Security: Protecting the Soft and Chewy Center”, Alec Main; <http://www.stsc.hill.af.mil/crosstalk/2005/10/0510Main.html>

“The Best-Laid Plans: A Cautionary Tale for Developers”, Lauren Weinstein; <http://www.csl.sri.com/users/neumann/insiderisks05.html#184>

©2012 Irdeto, All Rights Reserved.
 This document and the information contained herein is the subject of copyright and intellectual property rights under international convention. All rights reserved. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form by any means, electronic, mechanical or optical, in whole or in part, without the prior written permission of Irdeto. All non-Irdeto company names, product names, and service names mentioned are used for identification purposes only and may be the registered trademarks, trademarks, or service marks of their respective owners. All information is without participation, authorization, or endorsement of the other party.