
# Threat Driven Security

## Ensuring effectiveness through a systematic approach to protection design

**Whitepaper**

September 2015

# Threat Driven Security

## Introduction

Have you ever seen a young toddler receive a toy tool box filled with tools? One of the first things they realize is that they have a hammer, and that this hammer is powerful - in the blink of an eye everything around them looks like a nail, even toes. This creative use of their new tool is wonderful to observe and is even encouraged. Unfortunately, this is all too often the story of how security tools are used throughout the industry.

Cryptography is a prime example of a "hammer" applied too liberally to cast security problems as "nails". It is a fantastic tool when implemented properly and applied to address appropriate problems. However, security practitioners frequently mistake it for a silver bullet—a solution that magically solves all security issues. For example, let us consider a password-protected encrypted zip file. If the password chosen is robust, the data within the file is indeed secure. Yet, a naïve understanding of security may result in someone emailing the encrypted zip file with the password provided in-the-clear within the email. The use of encryption simply moves the threat from unzipping the file to obtaining the password sent in-the-clear through e-mail. Though contrived, this example highlights why a sound understanding of system security is needed so that we do not hit the wrong nail.

The tool box approach to security is the default approach for someone who does not have an appropriate understanding of how to address his system threats. Directly applying security technologies without clearly identifying the threats faced by a system is inefficient and frequently ineffective. Security problems must be considered at the system level by both systems implementers, and security solution providers. Implementers need this understanding so that they can apply security technologies appropriately while providers need to understand these threats so that they can produce meaningful security solutions.

Microsemi® straddles this division; on one hand we are a security solutions provider. However, we are often consulted for system-level protection design services as well. This document captures our best-practices, refined over more than a decade of producing efficient, effective mitigations for system-level threats.

## Overview

Threat-driven security is a systematic system-level approach that is driven by a clear understanding of the security need. It is not an arbitrary application of security technologies based on their perceived effectiveness or hype. Using this systematic approach, the strength of a protection is easily gauged through simple identification of the weakest link in the design.

A robust threat-driven protection is intended to make the investment required of a would-be adversary large enough that exploit development is impractical. A properly constructed threat analysis is tree-shaped, and ensures that the branches of the tree conclude with residual vulnerabilities that will remain in the system. Mitigations are selected until the residual vulnerabilities that remain are deemed acceptable. A determined adversary with sufficient resources can compromise any protection. However, most adversaries do not have unlimited resources, are time constrained, and typically look for the easiest prey. A careful evaluation of the remaining vulnerabilities of a protection must be used to make a risk-based assessment of the security solution quality.

Figure 1 shows the general outline of a threat tree produced during threat analysis.

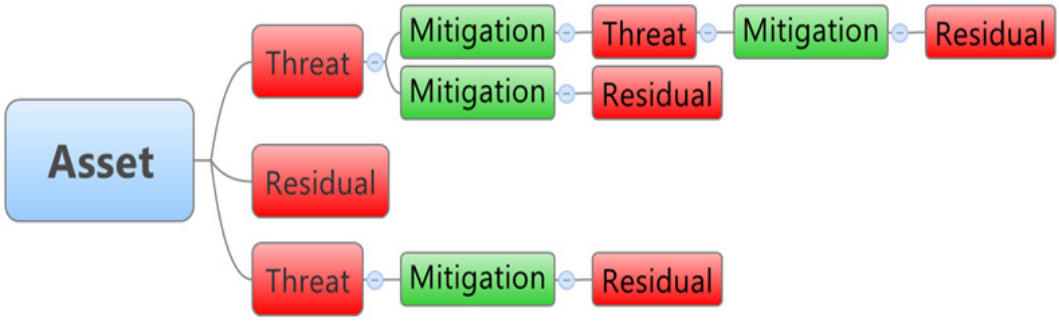


***Figure 1* • Threat Tree Analysis Approach**

The following steps describe the method to perform threat driven security analysis:

1. Identify what asset needs protection and why it needs to be protected
2. Systematically identify the threats that could compromise the asset
3. Identify mitigations for all material threats
4. Identify ways the mitigations could be circumvented—these are new threats
5. Repeat steps 3 and 4 until the remaining threats are deemed acceptable
6. Analyze the set of remaining vulnerabilities, to gauge the effectiveness of the protection design

The first step in designing a protection is to determine what needs to be protected and to establish a clear understanding of why it needs to be protected. It is all too easy to say that "we need security" without understanding what needs to be protected and to what types of threats the target might be exposed. When armed with this understanding we can begin to recognize the relevant threats that could expose, exploit, or compromise the asset.

As shown in Figure 1, we begin by enumerating the threats that could compromise the identified asset with three threats (threat #1, residual threat and threat #3) shown in the example. Each threat is individually analyzed to gauge the likelihood of occurrence—effectively categorizing each threat by risk. Based on the associated risk for the threat, it will either be deemed acceptable or in need of mitigation. Mitigations are identified for each of the high-risk threats. It is important to note that mitigations become part of the system, and are therefore themselves subject to attack. Hence the threat analysis and mitigation process is applied recursively. This process repeats until all remaining threats have an acceptable level of risk.

The remaining acceptable threats are the residual vulnerabilities associated with the protection. The strength of the protection is as strong as the weakest branch of the tree. A protection with a weak branch is like building a fortress with three impenetrable walls and one wall made of paper. The threat tree helps to quickly identify security weaknesses in a design, and encourages a holistic, system-level approach to security.

Using this threat-driven approach, a security engineer can make educated decisions about which security technologies to leverage in a protection design and can avoid the use of perceived "silver-bullet" security approaches.

# Embedded Software Threat Tree

To help solidify this abstract discussion of threats, let us turn to a simple example for intellectual property (IP) within embedded software. System-on-a-chip (SoC) devices are gaining widespread adoption in embedded systems giving their immense flexibility and computational capability. These SoC devices span a wide range of processor architectures and often contain embedded memory elements, co-processors and even field programmable gate array (FPGA) fabric—the Microsemi SmartFusion®2 SoC FPGA is such a device. Embedded systems leveraging these devices are often concerned with IP protection of the software running on these chips.

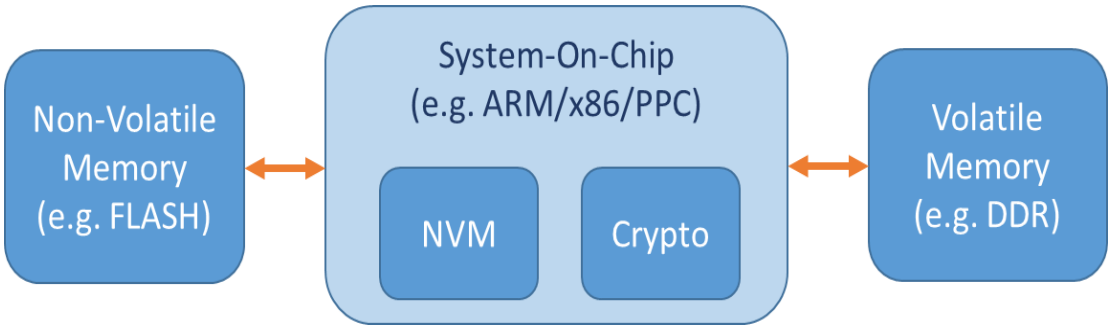


***Figure 2* • Example System-on-Chip Block Diagram**

Consider the generalized SoC architecture shown in Figure 2. In this example system, an SoC executes software that is stored at rest in non-volatile memory using external volatile memory for run-time operations. Following the procedure outlined above, we first identify the motivation for the protection to prevent the theft of IP present in the run-time software for the system.

It is helpful to consider attacks based on the state of data at the time of the attack:

- data-at-rest (DAR)
- data-in-use (DIU)
- data-in-transit (DIT)

By following this approach, we recognize a number of threats on the embedded software as illustrated in Figure 3. If this software is stored in the system un-encrypted, an adversary must only dump memory contents or capture the software at run-time with bus probing. As a mitigation against these attacks, we apply encryption to the boot image.


***Figure 3* • Simplified Threat Tree for Embedded Software**

With this mitigation in-place, additional attacks are identified including a brute-force password search, side-channel analysis to recover key material, or even probing of external interfaces not protected by encryption.

This simple example highlights the threat-driven approach to system protection design—a formal threat analysis is more rigorous and involves careful consideration of system design specifics. Microsemi has extensive experience designing threat protections for a wide range of systems and uses this expertise to drive product requirements to help address the needs of security practitioners.

# General Threats and Mitigations

Having established the analysis framework for protection design, we now consider some typical threats faced by systems, some mitigations for these threats, and how Microsemi solutions can help.

***Table 1 •*** **System Level Threats and Mitigations**

| | System Threats | Typical Mitigations | Microsemi Solutions |
|---|---|---|---|
| **Bus Probing** | SDRAM | Scrambling/Cryptography | EnforcIT® (SEMC) |
| | JTAG/SPI/Debug | Disables/Detectors | SmartFusion2/IGLOO2<br>EnforcIT (Security Monitor) |
| | Memory Dumping IC Lifting | Scrambling/Cryptography | TRRUST-Stor<br>SmartFusion2/IGLOO2<br>EnforcIT (Crypto Suite) |
| | Other | Scrambling/Cryptography | MACsec/Intellisec™<br>SmartFusion2/IGLOO2<br>EnforcIT (Crypto Suite) |
| **Network Exploits** | Implementation Bugs (e.g. Buffer Overflow) | Operating System Security<br>Static Analysis Tools<br>Procedural Changes | – |
| | Protocol | Open Standards | – |
| | Denial of Service (DoS) | Redundancy | – |
| | Snooping/Sniffing | Cryptography | MACsec/Intellisec<br>WhiteboxCRYPTO™<br>EnforcIT (Crypto Suite) |
| | Spoofing | Cryptography | |
| **Social** | Credentials Weak Passwords | Multi-factor Authentication | – |
| | Insider Leakage | Access control | – |
| | Design information | Access control | – |
| **Microscopy** | Optical | Detectors | – |
| | X-Ray | Detectors | – |

Table 1 shows a collection of system-level threats and mitigations. It is helpful to consider an example system when exploring the potential attacks that can be performed. In general, the system-level threats are leveraged at the network and board level as opposed to on a specific integrated circuit (IC). A few of the most common exploits are shown in theTable 1:

- Bus probing is used to capture data or even stimulate various system interfaces.
- Remote network attacks can be used to gain unauthorized access to the system, to inspect data-in-transit, or even to forge external traffic.
- Social attacks represent a unique collection of methods that can give attackers access to data and systems, that are otherwise well secured—these attacks use social conventions and pressure to trick well-meaning employees into divulging sensitive information such as account names and passwords.
- Microscopy is used to gain a general understanding of a system design and layout.

Microsemi has a wide range of solutions that can help mitigate these threats as shown in the Table 1. These technologies are provided as system, silicon, and IP level solutions.

***Table 2 •*** **Hardware/IC Threats and Mitigations**

| | Hardware/IC Threats | Typical Mitigations | Microsemi Solutions |
|---|---|---|---|
| **Side-Channel** | Simple Power Analysis (SPA) | Side-Channel Resistant IP | SmartFusion2/IGLOO2 EnforcIT (HardAES) |
| | Differential Power Analysis (DPA) | Side-Channel Resistant IP | |
| **Glitching Fault-Injection** | Voltage | Detectors/Filters | Fault-Injection: SmartFusion2/IGLOO2 |
| | Clock | Detectors/Filters | |
| | Laser | Detectors | Clock: EnforcIT (Security Monitor) |
| | Focused Ion Beam | Integrity verification | |
| | Optical | Integrity verification | Voltage/Clock/Laser: Coming Soon |
| | UV Light | Redundancy | |
| **Probing** | Needle Probes | Active Mesh | SmartFusion2/IGLOO2 |
| | Electromagnetic | Detectors | |
| **Microscopy** | Scanning Electron Microscope | Reconfigurable logic | Reconfigurable Logic: SmartFusion2/IGLOO2 |
| | Optical | Reconfigurable logic | |
| **Debug IFs** | JTAG | Disables/Detectors | SmartFusion2/IGLOO2 EnforcIT (Security Monitor) |
| | Other | Disables/Detectors | |
| **Remanence** | SRAM/DRAM | – | – |
| **Cloning** | – | Physically Unclonable Function | SmartFusion2/IGLOO2 EnforcIT (PUF) |

Table 2 shows the next class of threats hardware level attacks with a specific focus on ICs. The attacks range in sophistication from reading information over debug interfaces to performing chip-level manipulations. In many cases, the attacks can be performed when the target device is at rest. This makes detection rather difficult. When the device is running, some attacks can still be performed while avoiding detection, for instance, side-channel analysis.

There are a number of mitigations that can be used to disable, detect, and determine many of the attacks shown in the Table 2. The SmartFusion2 SoC FPGA by Microsemi is a prime example of an IC that strives to address several of the attacks shown in Table 2. It provides developers with a robust root-of-trust for their systems and offers built-in security tools for leveraging that root to provide security for the system. Refer to the *SmartFusion2 and IGLOO2 Security and Reliability User Guide* for more information on security and reliability.

Table 3 on page 7 shows the final category of threats: those related to attacking software. Software attacks can be categorized into a few main areas:

- Attacks that involve reverse engineering software at run-time with the aid of a debugger or at rest with the aid of disassemblers
- Modification of applications at rest or run-time to evoke a specific behavior such as a bypass of license checks
- Situations in which code is directly extracted from a system and re-hosted without any reverse engineering or modification. In this manner, an adversary can steal an algorithm without knowing the detail of how it works

There are a number of techniques that can be used to mitigate these threats which include surgical manipulation of a program to introduce obfuscation techniques, just-in-time encryption and decryption, integrity verification, and hardware binding. Microsemi has robust capabilities related to software protection with WhiteboxCRYPTO, which protects software implementations of cryptography through advanced key, code obfuscations techniques, and CodeSEAL™, which performs object code modifications to instrument applications with run-time code protections.

***Table 3 •*** **Software Threats and Mitigations**

| Software Threats | Typical Mitigations | Microsemi Solutions |
|---|---|---|
| **Debuggers** | Data Obfuscation Debug-hook detectors | WhiteboxCRYPTO<br>CodeSEAL |
| **Static Analysis/Reverse Engineering (RE)** | Obfuscation JIT Encryption | WhiteboxCRYPTO<br>CodeSEAL |
| **Code Insertion/Modification** | Integrity verification | Secure Boot<br>CodeSEAL |
| **Code Lifting** | Hardware/Software Binding | Secure Boot<br>WhiteboxCRYPTO |

The threats discussed in this section provide a broad sampling of those encountered in systems and the corresponding approaches used to address these vulnerabilities. Using the threat-driven approach to security, we can take a pragmatic approach to securing our systems by mapping these threats to our assets and identifying the mitigations that can help satisfy our security objectives.

# Security Needs by Market Segment

The final perspective needed by security designers is an understanding of the security needs of various systems. A protection design should address the specific security needs of a system and avoid deploying security techniques that are unnecessary and do not appropriately address the security threats faced by the system.

It is helpful to establish a few general motivating factors behind a need for security. These generalized categories help frame our approach to protection designs.

## *Access*

All systems are concerned about ensuring that access controls remain in place. Without access to a system, it is difficult for an adversary to perform exploits. Typically the access control concern is focused on network access; however, there are situations in which physical access control is important (For example, a bank or a weapons system).

## *Privacy*

Many systems, often those involving personal identifiable information (PII), are concerned with ensuring privacy. With the countless data spills that have revealed such personal information, the need for privacy-driven security is steadily increasing.

## *Reliability*

Reliability is a concern for those needing to ensure the uptime of their systems is maintained. Financial markets may be concerned with avoiding retail or stock-market downtime. Communication markets may be concerned with ensuring reliable communication is maintained. Industrial markets are concerned with preventing a take-down of the electric grid. Whatever the primary reason, security is often essential to reliable system operation.

## *Safety*

For many markets, safety is the primary motivating factor for leveraging security techniques. For example, it is important to prevent an adversary from taking control of an aircraft, deploying an airbag while a vehicle is in motion, or disabling a pace maker.

### IP Theft

IP Theft is a principal motivating reason for system security. Theft takes various forms depending on the type of system, but typically an adversary is trying to obtain financial gain or avoid investing in IP development through illicit acquisition.

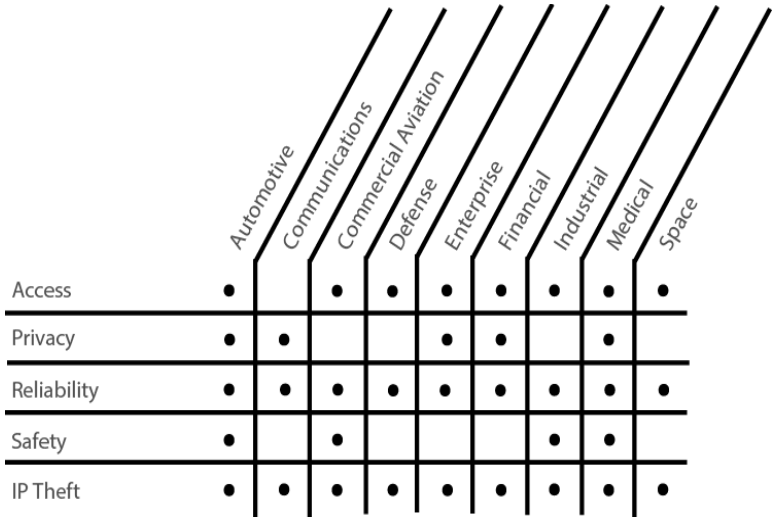


***Figure 4* • Reasons for Security by Market Type**

A general mapping of these security categories to various market segments is shown in Figure 4. The mapping is not absolute and is simply meant to help guide a protection design. There are safety reasons a defense system may employ in order to ensure security or access control concerns for communication systems; however, Table 3 on page 7 attempts to identify the primary motivating factors for each of the markets.

## Conclusion

Practicing a threat-driven approach to security provides a risk-based framework to identify the assets that need protection, the threats of concern for these assets, and the security solutions that should be used to protect the assets. Microsemi has a pedigree designing threat-driven protections and developing security products and solutions that help address the concerns faced by systems in need of protection. Contact Microsemi for help securing your system.


**Microsemi Corporate Headquarters**
One Enterprise, Aliso Viejo,
CA 92656 USA

**Within the USA**: +1 (800) 713-4113
**Outside the USA**: +1 (949) 380-6100
**Sales**: +1 (949) 380-6136
**Fax**: +1 (949) 215-4996

**E-mail:** sales.support@microsemi.com



Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,600 employees globally. Learn more at **www.microsemi.com**.