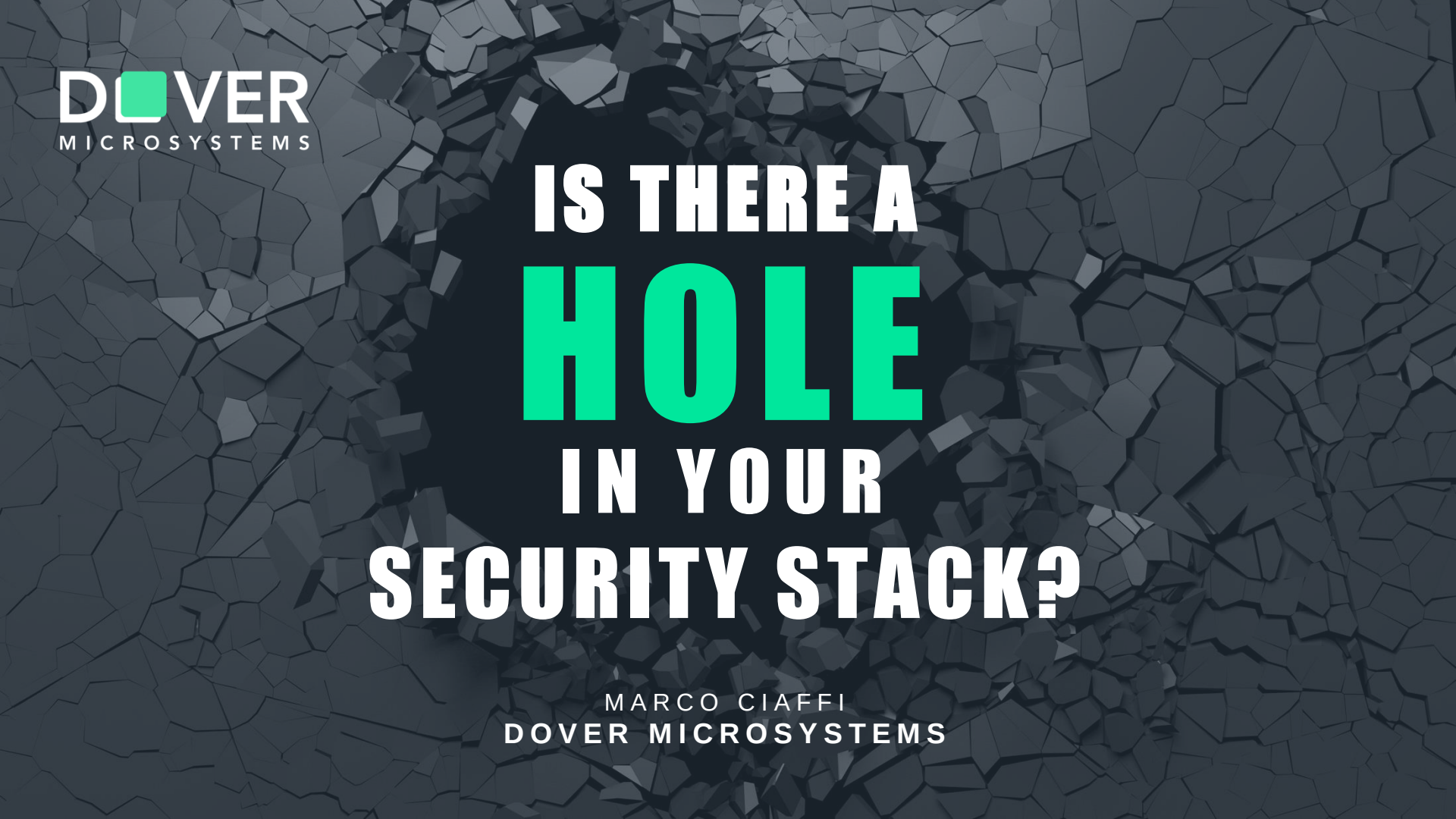
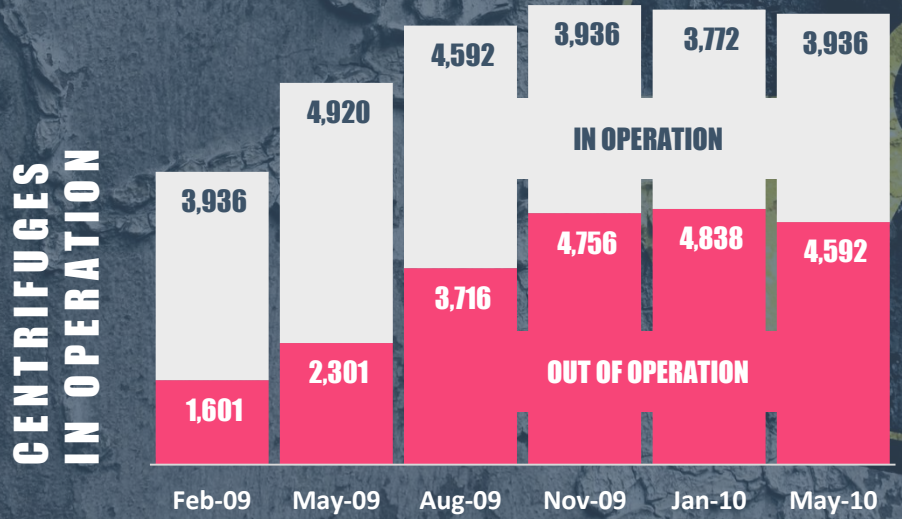

DOVER
MICROSYSTEMS
IS THERE A
HOLE
IN YOUR
SECURITY STACK?
MARCO CIAFFI
DOVER MICROSYSTEMS

I understand the difference in destruction is dramatic, but this has a whiff of August 1945. Someone just used a new weapon, and this weapon will not be put back into the box.

Michael Haden
FORMER DIRECTOR OF THE NSA & CIA

2009 STUXNET ATTACK



Sources: IAEA, ISIS, FAS, World Nuclear Association, FT research

OSI MODEL

High-level APIs

Translation of data between a networking service & an application

Manage a continuous exchange of information between two nodes

Reliable transmission of data segments between points on a network

Address, routine, and traffic control for a multi-node network

Reliable transmission of data frames between two nodes connected by a physical layer

Transmission and reception of raw bit streams over a physical medium

OUR INSPIRATION

TODAY'S CYBERSECURITY STACK IS INCOMPLETE

SOFTWARE

APPLICATION Credentials, Sanitization	Manages credentials for authorized users and runs sanitation routines that check SQL queries.
KERNEL Operating System, Intrusions, Virus Scans	Scans for signatures, detects intrusions, and prevents unauthorized access to a network.
ENCRYPTION Communications, Data-in-Motion	Prevents data theft by converting data into code that can only be decrypted using an authorized key.
COMPARTMENTALIZATION Hypervisors, Zones, TEE	Isolates critical pieces of software in a sandbox so that they cannot be corrupted.

HARDWARE

ROOT OF TRUST Keys, Secure Boot, Crypto	Validates all the hardware and software on the system at boot time.
PHYSICAL Tamper, Supply Chain, Rad-hard, Fault Tolerant	Ensures unauthorized personnel cannot touch or tamper with the system in the real world.

EXAMPLE VENDORS

McAfee, Symantec, Malwarebytes, FireEye, Rambus, Marvell, Fortanix, 0x5 HEX-Five Security, intel, arm

Rambus, inside secure, maxim integrated, Teledyne Technologies, Secure Thingz

BUT THE TOP LAYERS ARE EXTREMELY VULNERABLE

Highly Vulnerable to Attack

SOFTWARE

APPLICATION
Credentials, Sanitization

Attacks against security software & privilege escalation. Install Spectre / Meltdown agents.

KERNEL
Operating System, Intrusions, Virus Scans

Attack OS & privilege escalate or install malware.

ENCRYPTION
Communications, Data-in-Motion

Attack origination point and bypass call to encryption or send fake data.

COMPARTMENTALIZATION
Hypervisors, Zones, TEE

Attack compartments or hypervisor.

HARDWARE

ROOT OF TRUST
Keys, Secure Boot, Crypto

PHYSICAL
Tamper, Supply Chain, Rad-hard, Fault Tolerant

All software has bugs.

Attackers turn bugs into exploits.

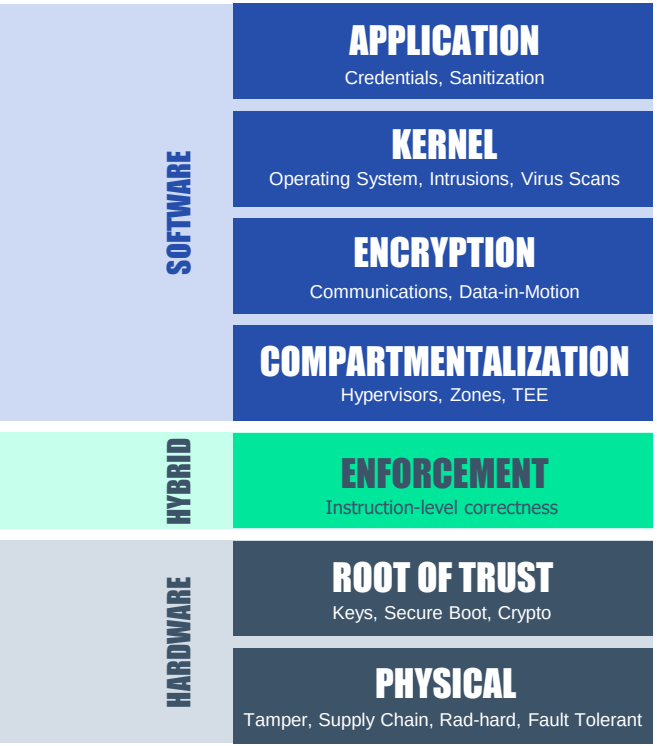
Most common attack scenario is buffer overflow:

FILLING THE HOLE WAS DARPA-HARD

$100 MILLION

CRASH PROGRAM

ENFORCEMENT PLUGS THE HOLE IN THE SECURITY STACK



With Enforcement your software layers are now secure

Inefficient signature-based scans are not necessary

Encryption cannot be bypassed

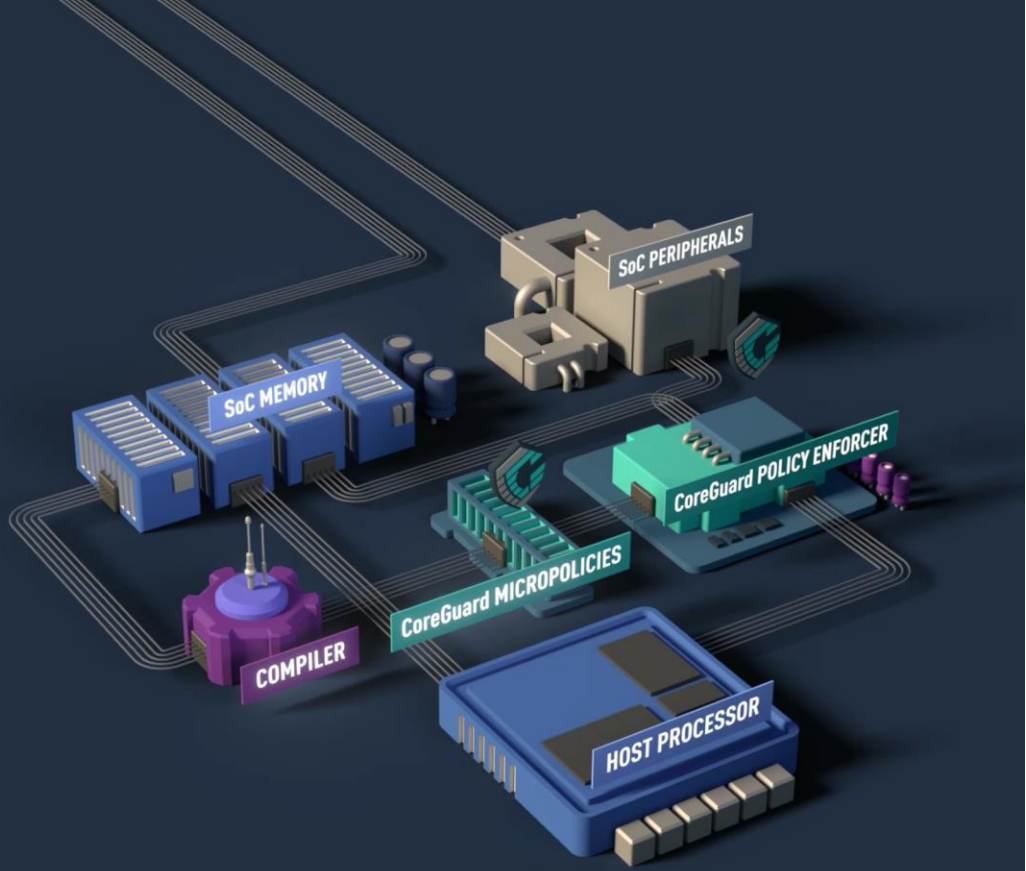
No zone vulnerabilities

Immunizes processors against entire classes of network-based attacks, including zero-days.

ENFORCEMENT

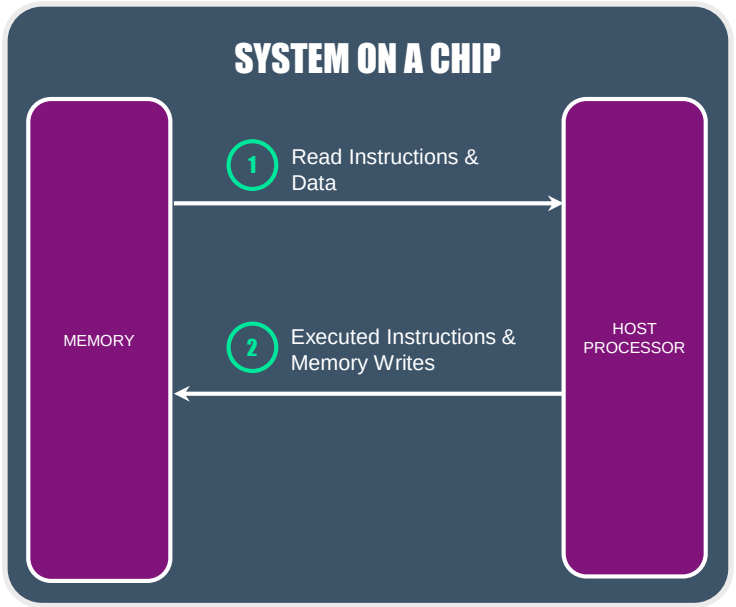
is the only way to **prevent the exploitation of software vulnerabilities**

HOW DOES ENFORCEMENT WORK?



PROCESSORS HAVE NO AWARENESS FOR SECURITY

Conventional processors are optimized for size and speed, **not security.**



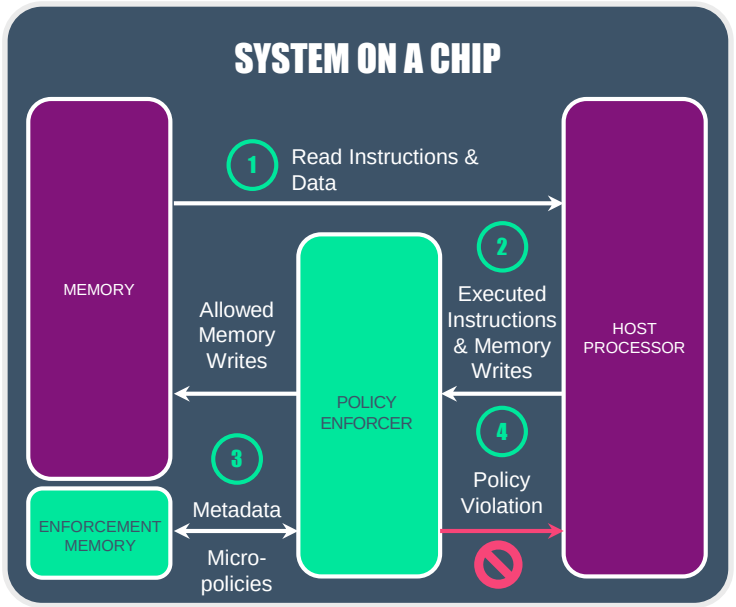
1. Read from memory **instructions** to be executed **and data** to be processed

2. The host processor executes instructions and **writes data back** to memory

THEY
BLINDLY
DO WHATEVER THEY ARE TOLD

TODAY'S PROCESSORS NEED A BODYGUARD

Enforcement solutions **monitor every instruction** executed by the host processor to ensure it **only does what it is meant to do.**



1. Read from memory **instructions** to be executed **and data** to be processed
2. Host processor **sends** the executed instructions and writes **to the Policy Enforcer**
3. Policy Enforcer **crosschecks captured metadata** for every instruction, **against a set of micropolicies**
4. If an instruction violates a micropolicy, the Policy Enforcer **stops it from writing any data back to memory**

AT THE CORE OF ENFORCEMENT

Enforcement
HARDWARE

Rules
MICROPOLICIES

Information
METADATA

ENFORCES SECURITY, SAFETY & PRIVACY

Micropolicies are designed to stop entire classes of attacks, including buffer overflows, code injection, data exfiltration, and even safety violations.

FOCUS	MICROPOLICY EXAMPLES		
SECURITY	HEAP PROTECTION	STACK PROTECTION	RWX (READ, WRITE, EXECUTE)
	GLOBALS PROTECTION	DATA TYPE ENFORCEMENT	PROCEDURE ENFORCEMENT
	CONTROL FLOW INTEGRITY	FINE-GRAINED ACCESS CONTROL	SANDBOX
	COMPARTMENTALIZATION	CODE PROTECTION	RESOURCE MANAGEMENT
PRIVACY	INFORMATION FLOW CONTROL	MULTI-LEVEL SECURITY	DATA EXFILTRATION PREVENTION
SAFETY	MEDICAL / AUTOMOTIVE	AI OPEN-LOOP PREVENTION	FINITE STATE MACHINE ENFORCEMENT

HEAP SAFETY MICROPOLICY

Example security micropolicy in action

GOAL

Provide spatial and temporal safety

Stop buffer overflow attacks

METHOD

Give each pointer a unique “color” or tag

Color memory slots with this tag on allocation

Recolor on Free

MICROPOLICY

```
storeGrp(addr == color, mem == color -> mem = color)
```

Metadata on pointer to: x y z

Metadata on address to STORE

Memory stays the same

x y z

```
x = malloc(2);

x[0] = 0x09;

y = malloc(5);

y[3] = 0x04;

z = malloc(3);

z[1] = 0x01;

x[2] = 0xbad;
//FAIL
```

Payload data in host core
0x09
0x08
0x07
0x06
0x05
0x04
0x03
0x02
0x01
0x00
Metadata

THE ENFORCEMENT ADVANTAGE

IMMUNIZE PROCESSORS

Stops entire classes of network-based attacks

DEFENSE AGAINST BUGS

Prevents the exploitation of software vulnerabilities

SECURITY IN SILICON

Cannot be subverted over the network

REAL-TIME PROTECTION

Blocks attacks in real-time, before any damage can be done

CUSTOMIZABLE & UPDATABLE

Micropolicies can be customized to application and securely updated as needed

SECURITY STACK PROTECTION

Protects other layers of the security stack and eliminates costly signature-based scans

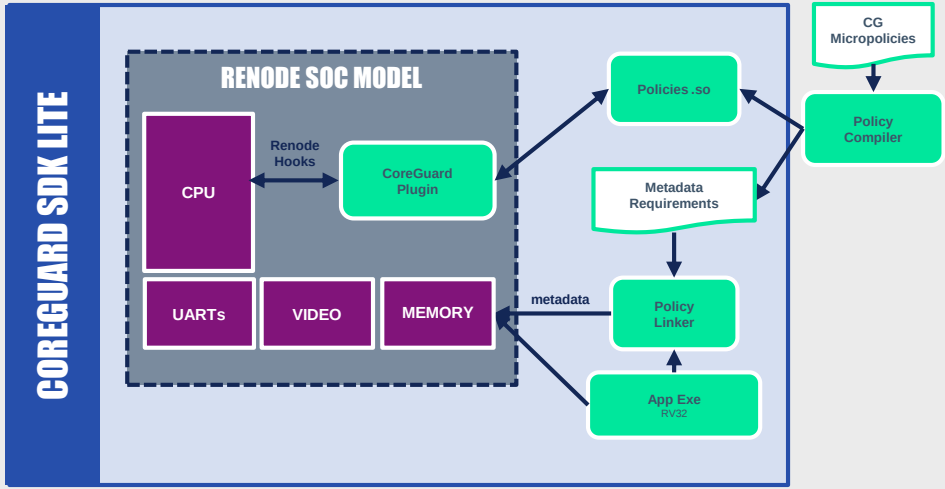
SEE ENFORCEMENT IN ACTION

Download our free SDK Lite to see how CoreGuard®, the first Enforcement solution for embedded systems, can stop entire classes of network-based attacks.

RENODE™

Delivered as a virtual machine with sample applications and debugger—includes a curated set of micropolicies and a subset of developer tools

- SoC has AP CPU + peripherals
- CoreGuard is entirely simulated in C# and C++
  - Policy code runs on host (e.g. X86)
- Via a Renode Plugin
  - Registers BlockBeginHook and BlockEndHook Renode hooks
  - Hooks call generated C++ code



COMPLETE YOUR STACK WITH ENFORCEMENT

SOFTWARE

APPLICATION
Credentials, Sanitization

KERNEL
Operating System, Intrusions, Virus Scans

ENCRYPTION
Communications, Data-in-Motion

COMPARTMENTALIZATION
Hypervisors, Zones, TEE

HYBRID

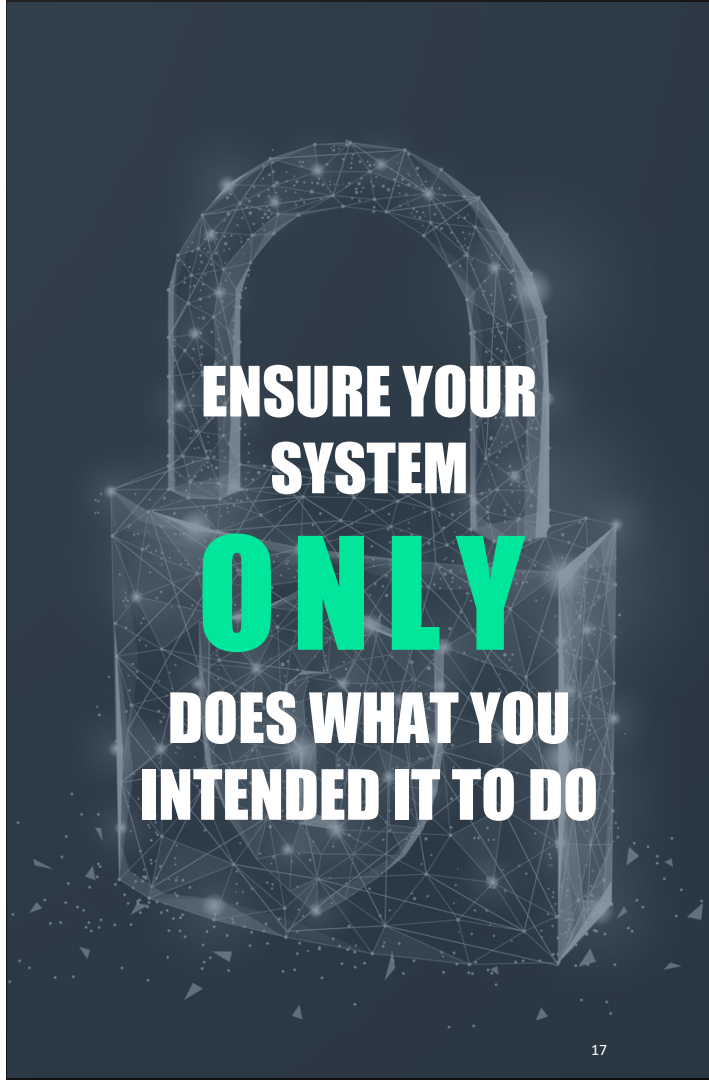
ENFORCEMENT
Instruction-level correctness

HARDWARE

ROOT OF TRUST
Keys, Secure Boot, Crypto

PHYSICAL
Tamper, Supply Chain, Rad-hard, Fault Tolerant

PROTECT THE OTHER LAYERS OF YOUR STACK AGAINST NETWORK-BASED ATTACKS



THANK YOU

MARCO CIAFFI

Co-Founder & VP of Engineering
Dover Microsystems

marco@dovermicrosystems.com
www.dovermicrosystems.com