Opportunity

Keys to Success

Jothy Rosenberg, PhD
Founder & CEO

RISC-V Summit
December 3-6, 2018
Santa Clara, CA

RISC-V

# IF WE GET RISC-V SECURITY RIGHT

IT WILL BECOME THE DOMINANT PROCESSOR IN THE $470B IoT MARKET

Demand for Security

The Security We Need

DOVER MICROSYSTEMS

Jothy Rosenberg, PhD
Founder & CEO

RISC-V Summit
December 3-6, 2018
Santa Clara, CA

# IF WE GET RISC-V SECURITY RIGHT

## IT WILL BECOME THE DOMINANT PROCESSOR IN THE $470B IoT MARKET

# Threat | Opportunity

IoT is extremely insecure and presents an enormous threat to us all

This is going to create an opportunity for RISC-V if we–the RISC-V community–can address this threat

I believe we can

# The IoT Market Will Be Huge

IoT will be 30B connected devices by 2021

IoT market will be a half a trillion dollars by then (says Forbes)

# When There Is No Security ... Dyn Happens

Many IoT devices have built-in passwords, or simple passwords that are widely known

Mirai attackers made short shrift of these openings and created a massive bot army

If bot armies can attack name brands and take down leading websites for half a day ...
... security will be demanded, regulated, or even litigated

# RISC-V Can Dominate IoT

But it will require innovation and collaboration in security to capture that position

Otherwise Arm will take it

# Demand for Security Is About to Explode

Biggest need is protecting the processor from being taken over via **Software Vulnerability**

Few know that it is possible to achieve this kind of immunity against network-based attack

But who knew 1000 songs in your pocket was a possibility?

# Big Vendors Have No Effective Offering

Not in the IoT space so not relevant to this discussion
No one but Intel can do the right thing on Intel

They say and believe TrustZone (compartmentalization) is all you need–it's not
Plus, only Arm can do this to Arm ($20M+ architecture license per core)

# RISC-V Can Respond

Clean new architecture

Expandable ISA

Strong community interest in security

Open-source communities get security right (AES encryption)

Best of all we all have a

architecture license!

# The Security IoT Needs

# Compartmentalization

It prevents interference in data (or code) memory of one thread or process by another

Arm's  motivation was Android getting corrupted by malware in apps

Advanced processors and operating systems have had this for decades

RISC-V already has good compartmentalization capabilities

# Communications Security

The term "secure processor" is used to describe cryptography, designed to secure communications (data in motion)

Communications security is a relatively solved problem

Strong authentication, key management and crypto (commodity HW IP blocks or software)

Communications security prevents a "man in the middle" attack

RISC-V has strong communications security technology available to it

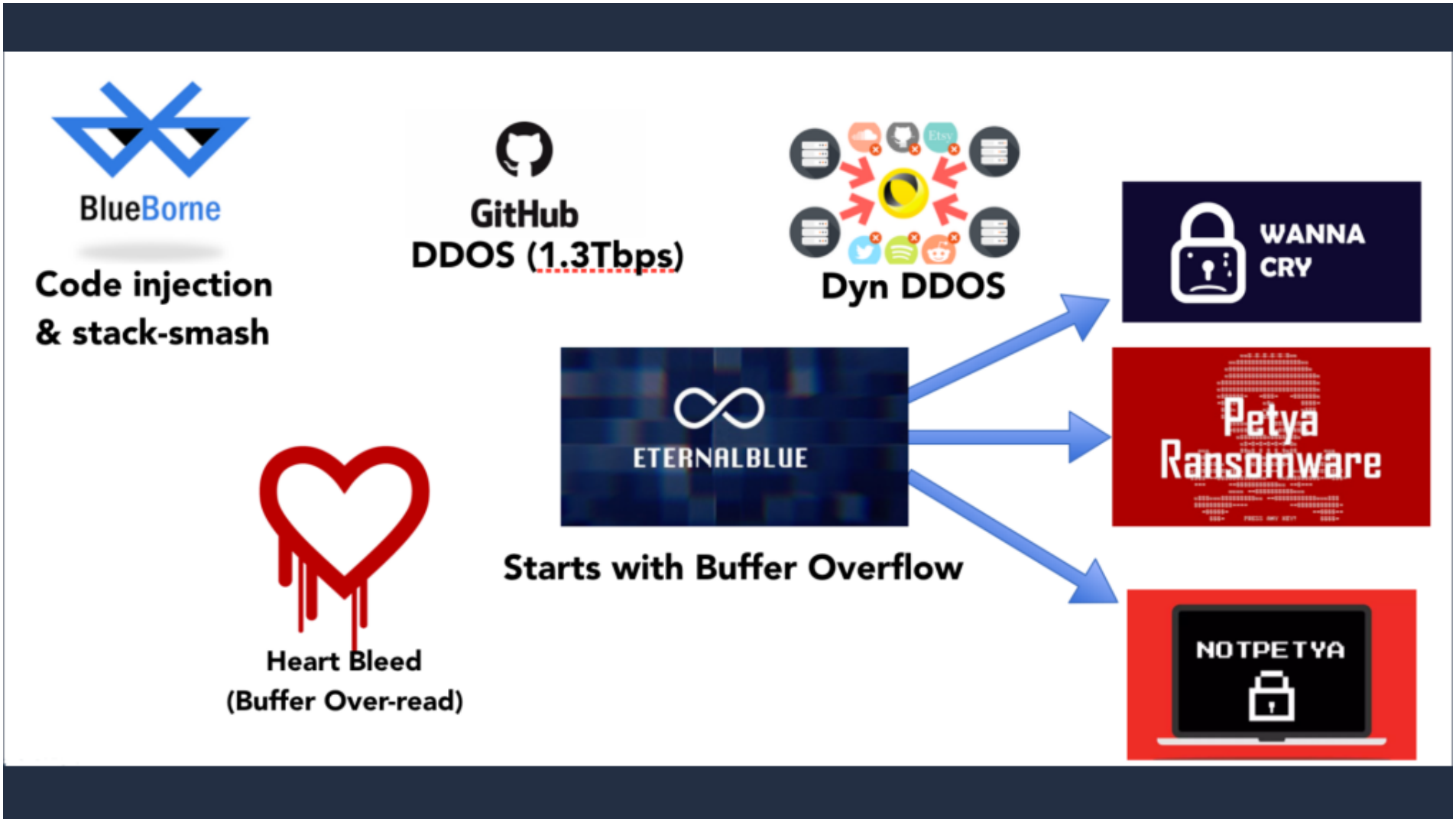
# Computing Security

Compartmentalization and communications security are not enough:

- Compartmentalization does not prevent network-based attacks because the trusted compartment runs complex code with exploitable bugs
- Communications security does not prevent data exfiltration because a determined hacker will bypass encryption of outgoing data

Computing security protects both compartmentalization and communications security from attacks (like buffer overflow) making them stronger and work correctly

Destructive and expensive network-based attacks are using buffer overflows to win


BlueBorne
Code injection
& stack-smash
GitHub
DDOS (1.3Tbps)
Dyn DDOS
WANNA
CRY
ETERNALBLUE
Petya
Ransomware
Starts with Buffer Overflow
NOTPETYA
Heart Bleed
(Buffer Over-read)

# Computing Security

Protecting the computing engine (processor) from being overtaken by cyberattacks that arrive via a network connection and then exploit software vulnerabilities to take control of the processor

# A DARPA-hard Problem

DARPA-hard Där,pä-härd
*adjective*
1 a problem that no one else can or will solve and many might think impossible

Motivated by Stuxnet, DARPA created $100M CRASH program where our research began

Recognition of a computing security gap: industry couldn't address it, so DARPA had to (try)

# The Dirty Little Secret

There are 15-50 bugs per 1,000 lines of delivered code

Over 111B lines of code are written per year

It takes 150M lines of code to run a Ford F-150

Today's processors are helpless to stop cyberattacks because of these bugs

# It's the Hardware (stupid)

The foundation on which computing security is built must be hardware

Network attackers cannot re-wire our hardware

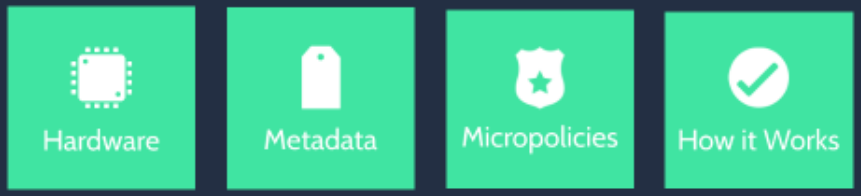
But the solution should not be 100% hardware– too brittle

Hybrid solution is unassailable yet flexible and updatable and can be made very secure

# The Secret Sauce

The secret sauce we developed at DARPA and Draper (and now at Dover) consists of:

- **Hardware** that becomes an unassailable enforcement mechanism
- **Metadata** that knows if an instruction is performing the right or wrong action
- **Micropolicies** (rules) that define which security, privacy, and safety properties are to be enforced



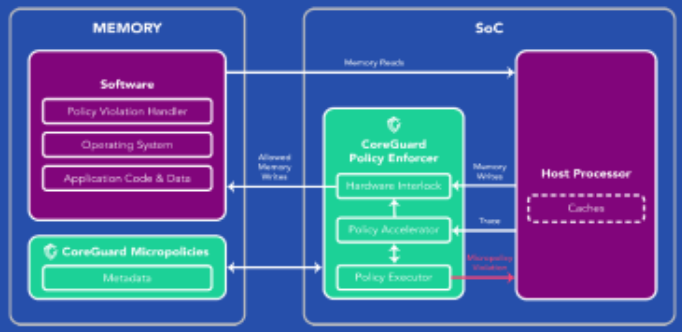
# Hardware: Enforcement

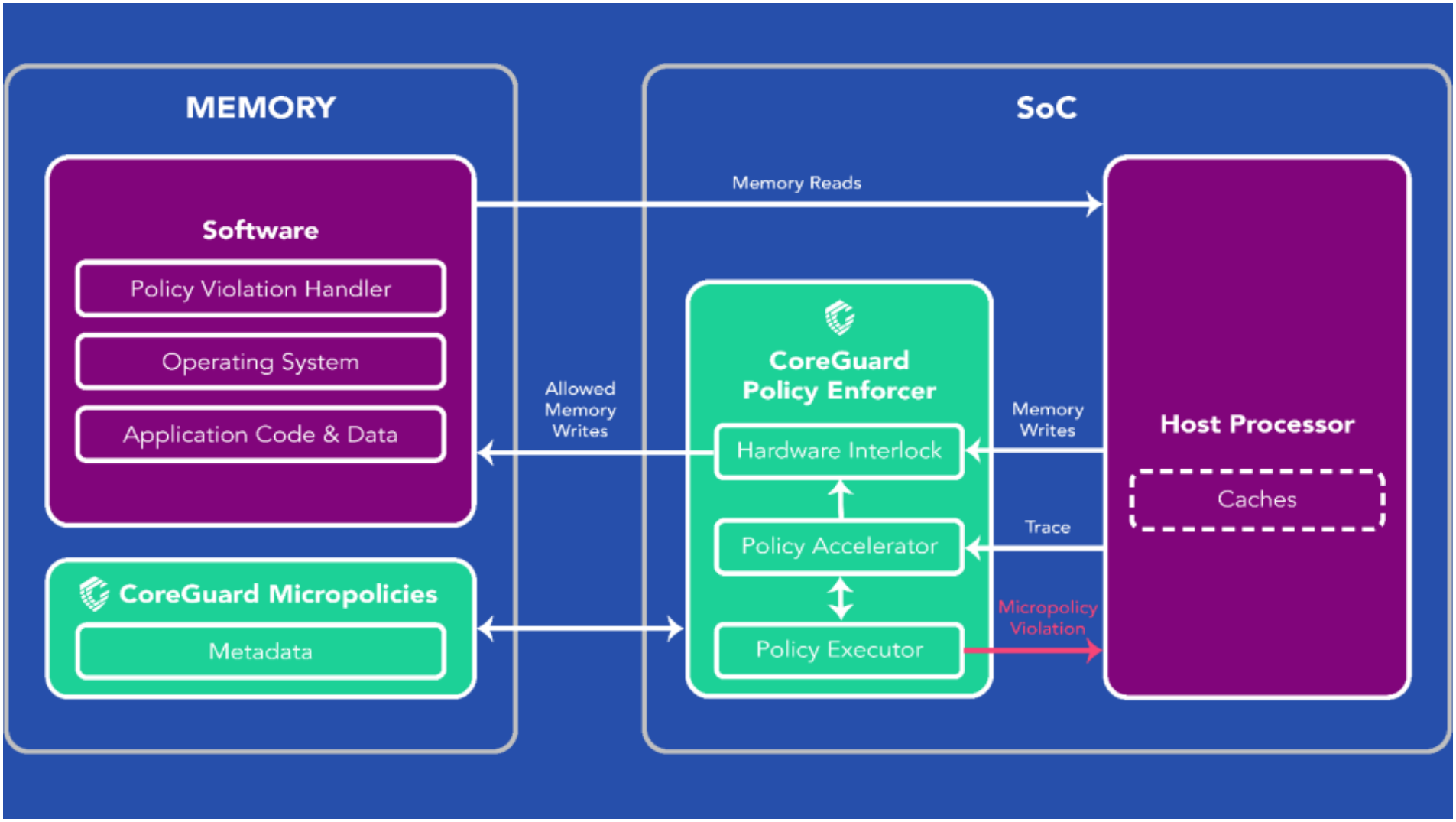
Extracts a set of trace signals from host processor

Monitors and evaluates every instruction in the host processor in real time

Provides a stall mechanism to stall the host when needed to evaluate micropolicies

Exceptions thrown when micropolicy violated



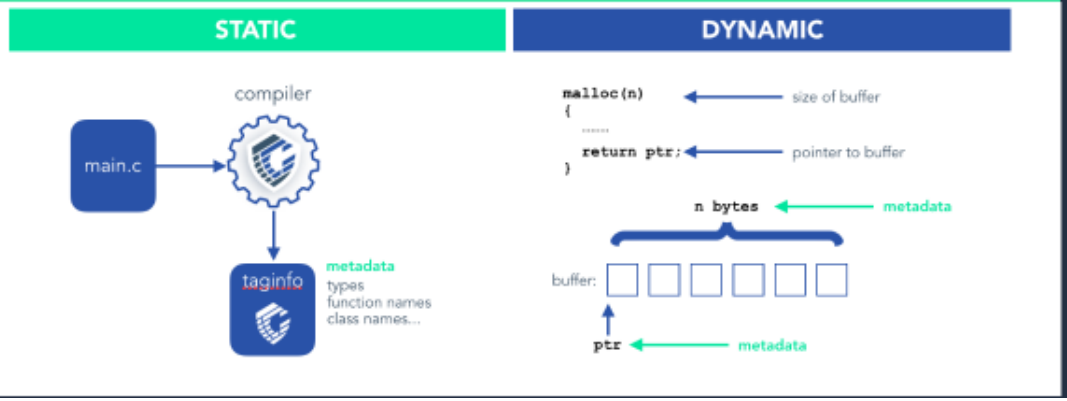

MEMORY
Software
Policy Violation Handler
Operating System
Application Code & Data
CoreGuard Micropolicies
Metadata
SoC
Memory Reads
Allowed Memory Writes
CoreGuard Policy Enforcer
Hardware Interlock
Policy Accelerator
Policy Executor
Memory Writes
Trace
Micropolicy Violation
Host Processor
Caches

# Metadata: Knowledge

To determine if an instruction is doing the right or wrong thing

Static data like the compiler's call graph so we can tell if control-flow is being hijacked

Dynamic data like from the inside of the malloc routine where it knows where the buffer starts / ends



# Micropolicies: Rules

Rules that define which security, privacy, and safety properties are to be enforced

Security rules determine if instruction is allowed

Privacy rules prevent exfiltration of confidential data

Safety rules guarantee only defined states are reached in pre-determined ways

| Focus | Micropolicy | Prevents |
|---|---|---|
| Security | Memory Safety | Buffer Overflow |
| | RWX (Read, Write, eXecute) | Code Injection |
| | Control Flow Integrity | Return Oriented Programming |
| | Compartmentalization | Rogue Processes |
| Privacy | Taint Tracking | SQL Injection & Privacy |
| Safety | Medical | Excessive insulin injection, defibrillation |
| | Automotive | Dangerous acceleration, braking, steering |
| | Industrial IoT | Untrusted data use in critical decision making |

# Heap Micropolicy

Stops buffer overflows

Buffer overflows are involved in over 80% of network-based cyberattacks

**Goal**

Provide spatial and temporal safety

Stop buffer overflow attacks

```
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 |

**Goal**

Provide spatial and temporal safety

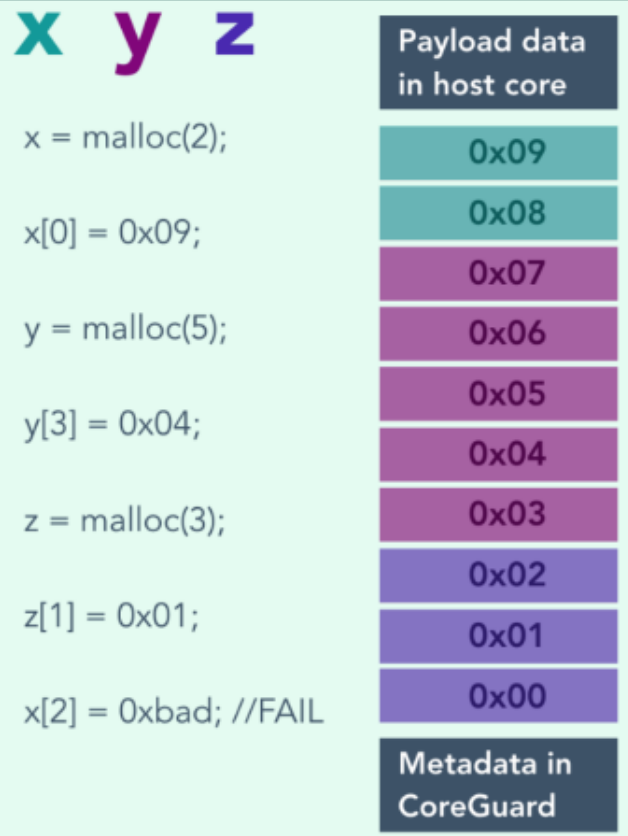
Stop buffer overflow attacks

**Method**

Give each pointer a unique "color" or tag

Color memory slots with this tag on allocation

Recolor on Free



**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 in CoreGuard

# Security & RISC-V's IoT Success Are Linked

Time to Stop These Attacks
We have to stop ransomware
We have to stop electric grid incursions
We have to stop breaches of consumer PoS data
We have to stop auto and medical system attacks
We have to prevent the Pearl Harbor cyber event

## Need Best in Class ...

Compartmentalization

Communications security

Computing security

# RISC-V Enables This

Strong security-minded open-source community

Open source: many eyes and hands working toward common goal–moves ecosystem faster

Will create safer IoT, enable organic IoT growth

This will cement RISC-V as the processor for the half trillion dollar IoT market

And it will fundamentally change the processor industry forever

# Jothy Rosenberg, PhD

Founder & CEO
jothy@dovermicrosystems.com
(617) 306-8121


DOVER
MICROSYSTEMS