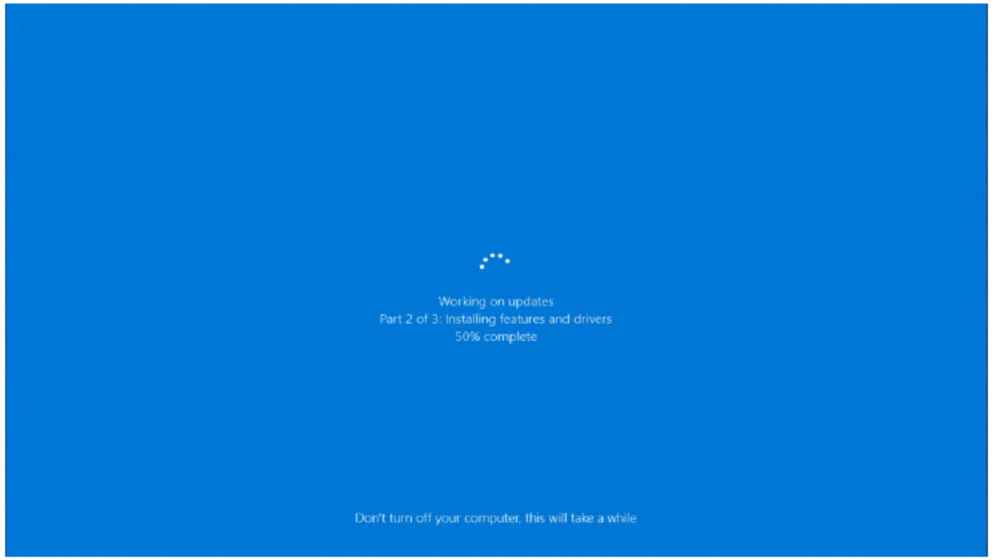
System Security Integrated Through Hardware and Firmware (SSITH)

Keith Rebello

May 2021

A Constant and Growing Threat

- $6 trillion in damages annually by 2021
- 23% of Americans impacted
- Ransomware attacks every 14 seconds
- Software patching reduces performance (20% in Microsoft Windows)



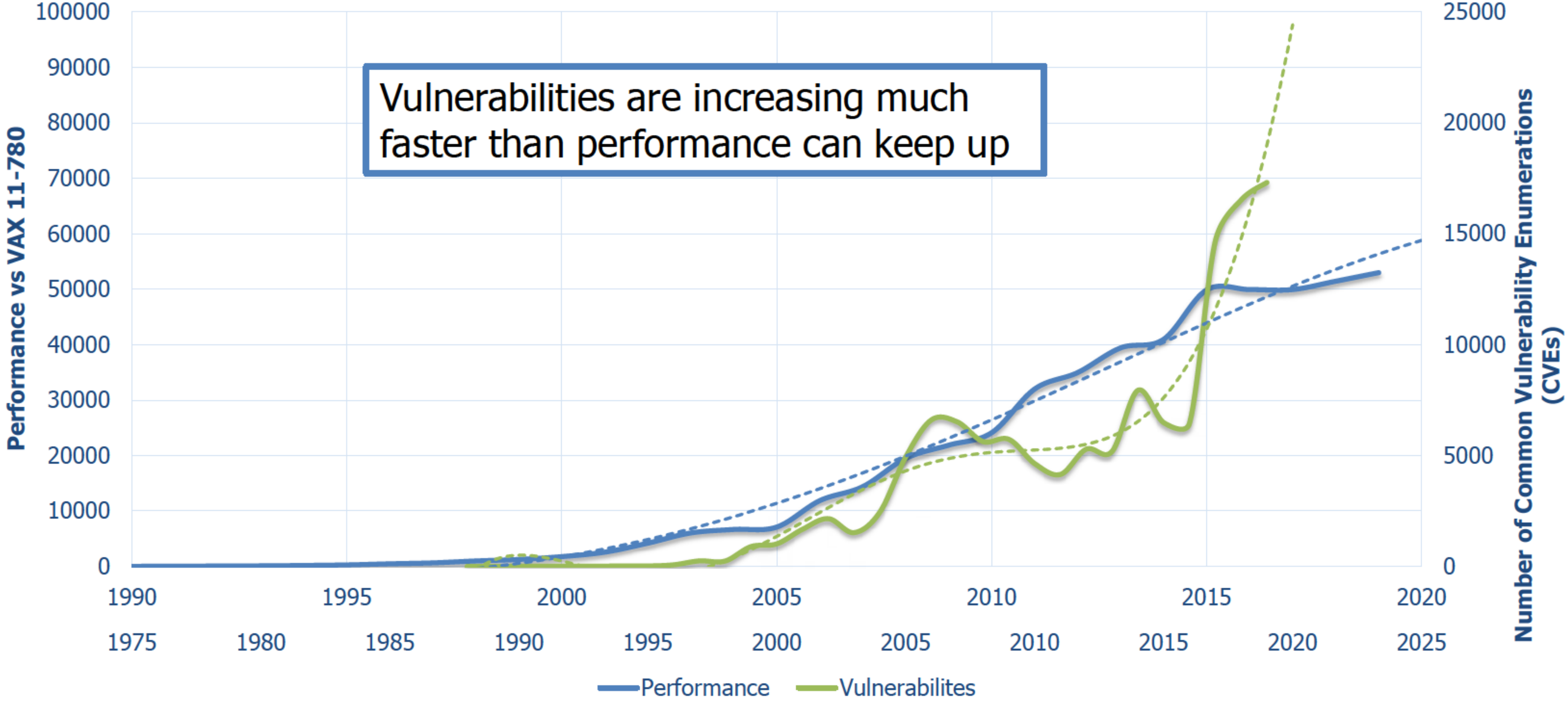
Working on updates
Part 2 of 3: Installing features and drivers
50% complete

Don't turn off your computer, this will take a while

Hacker Ne
hack: 15.2 MILLION UK records
Is your Data SAFE?
Hacking
Intrusion
Viruses
Cyber Threats
Theft
Attacks
Phishing
Security Breakdown
India lost $8bn to cyber crime
India is ranked fifth in e-commerce security breaches; over 14,000 websites hacked
STUXNET
THE HACK

Sources: shesecures.org, federal-technology.com, zseries.in, radware com, eenews.net, ZDNet.com

Performance and Security Trends



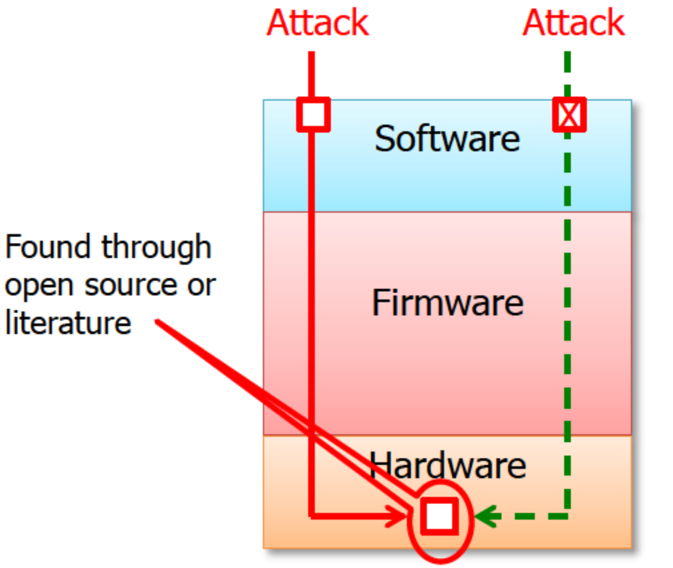
Vulnerabilities are increasing much faster than performance can keep up

DISTRIBUTION A: Approved for Public Release, Distribution Unlimited

Program Overview

Today: Patch and Pray

*2800 vulnerability instances
2800 software patches

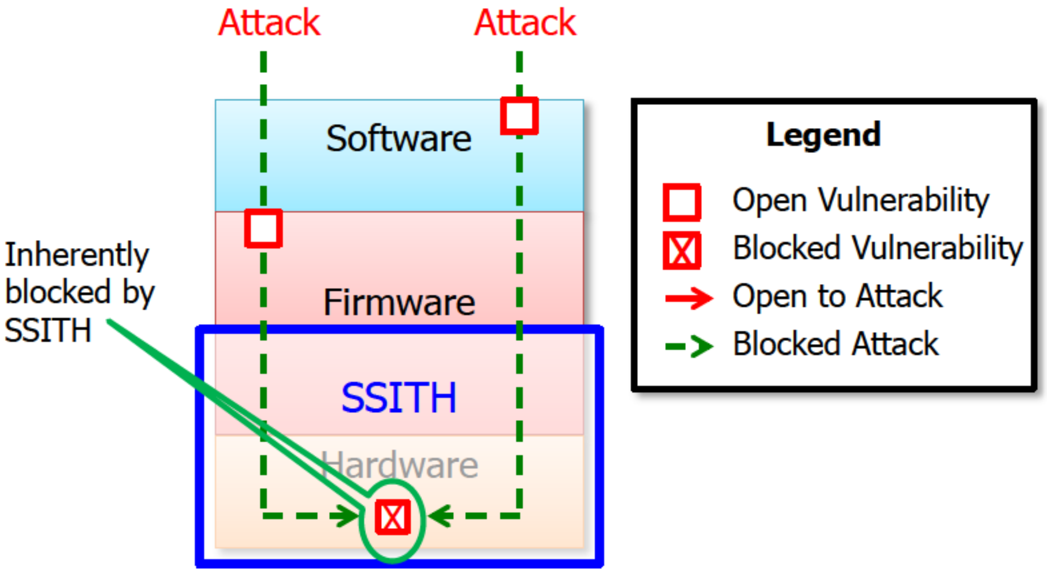


Hardware vulnerability class

2015 MITRE-recorded hardware vulnerabilities (CVE)

Future: SSITH

SSITH will protect against all 7 hardware classes



Hardware vulnerability class

*7 vulnerability classes
7 hardware solutions

SSITH addresses current and future hardware vulnerabilities at their source

Technical Requirements

- Close ALL seven classes of hardware weakness identified by NIST with acceptable overhead

CWE	Description	Phase
Buffer Errors	Reading or writing outside memory bounds of data structures.	1
Information Leakage	Unintentional information sharing or data transfers.	1
Permissions, Privileges, Access Control	Unauthorized access based on system authorizations.	1
Resource Management	Improper access to hardware resources (memory, CPU, I/O).	2
Numeric Errors	Improper calculation or conversion of numeric types.	2
Injection	Introduction of malicious code or data.	3
Hardware/SoC	Hardware-design and SoC implementation flaws.	3

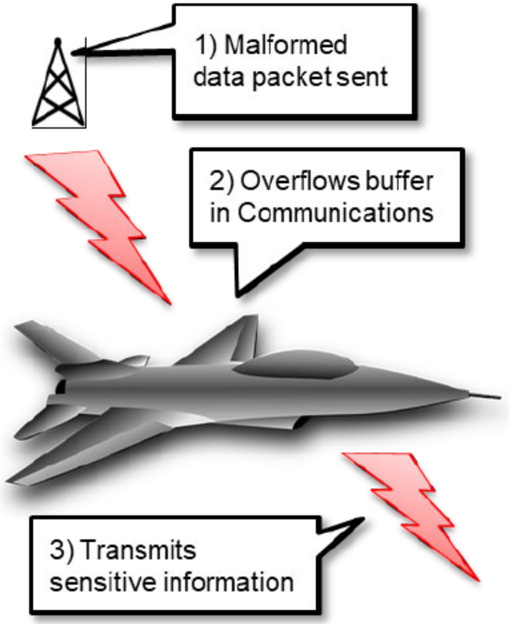
- Maintain software compatibility
  - Ideally no software changes
  - Recompilation is acceptable but discouraged
- TA1: Develop architectures which scale from simple 32 bit microcontrollers to 64 bit superscalar CPUs
- TA2: Develop methodologies and software tools to evaluate the performance of SSITH architectures
- TA3: Transition activities

SSITH protects against all classes of hardware weaknesses with minimal impact across architectures

SSITH CWE Example

Buffer Error CWE

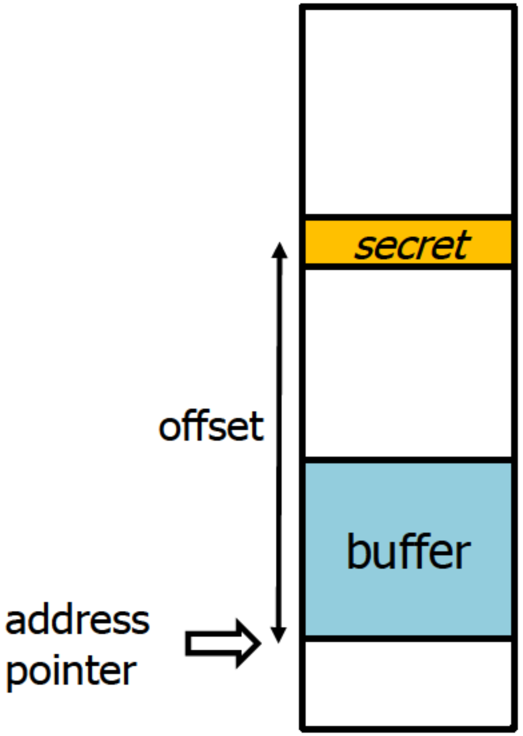
- There have been thousands of malware exploits based on “buffer overflow” attacks
  - New exploits happening all the time
  - Roughly one in five exploits uses buffer errors
    - (source: cvedetails.com across ~25 years)
- Buffer Errors occur when reading or writing beyond the bounds of a data structure
- Allows an attacker to read or write to arbitrary memory locations, enabling them to:
  - Obtain potentially sensitive information from memory
  - Cause memory corruption and/or crash the application
  - Execute arbitrary code on the target system.
- Current hardware architectures have no protections and will perform the read or write operation



1) Malformed data packet sent

2) Overflows buffer in Communications

3) Transmits sensitive information



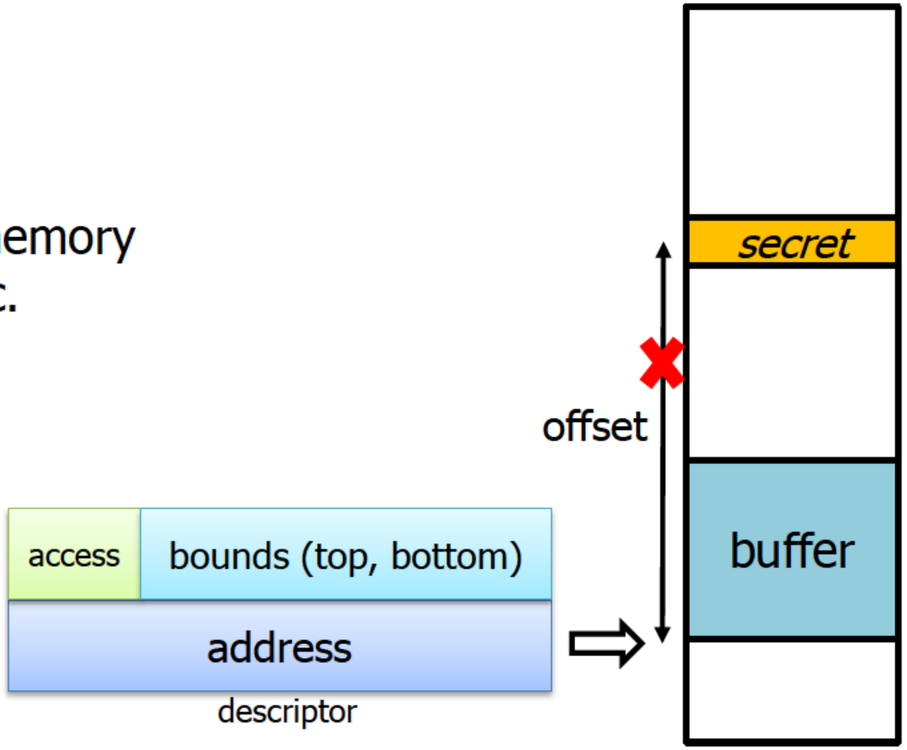
SSITH CWE Example (Continued)

SSITH Tagging

- Address pointers are replaced or modified by metadata that describes the object being addressed – such as:
  - Base address in memory
  - Object size limits (bounds)
  - Access permissions
  - Code or data
- The hardware pipeline is modified to check that the requested memory access is allowed based on object bounds, type, permissions, etc.

Tagging Variations

- All of the SSITH TA1 performers are using a variation of tagging
- Simple tagging may just be 2 bits indicating pointer to data or code
- Descriptors or "capabilities" can encode bounds, permissions, etc.
- Metadata can be encoded or contained in tables accessed by an index
- Tagging is useful for multiple CWEs, e.g. access control



SSITH tagging mechanism will eliminate Buffer Error weakness forever

Program Impact

- SSITH addresses current and future hardware vulnerabilities at their source
  - Software patches only needed for software bugs and weaknesses
  - Requirement to "patch" the hardware will be rare as SSITH covers all known classes of weakness
- Currently, state-of-the-art commercial approaches don't meet DoD requirements for hardware security across a broad range of applications and SoCs
- SSITH can be incrementally added to DoD SoC applications
  - Reduced re-certification overheads
- SSITH approach includes flexibility to scale from very simple embedded microcontrollers to high-performance superscalar processors

Sources: mirror.co.uk, defense-update.com

SSITH technology can be adapted to diverse application needs

SSITH Program

SSITH Program Overview

- Program started in December 2017
- Currently in final Phase 3 through October 2021
- SSITH ASIC effort will extend through October 2022

Program Manager: Keith Rebello

Program Performers

- Lockheed Martin: HARD Parallel Security Co-Processor
- SRI/Cambridge/Arm: CHERI Capabilities
- Agita Labs: Morpheus Secure Cloud
- Galois: BESSPIN System Security Evaluation

SSITH ASIC

- In planning
- ARM-based multicore ASIC demonstration board

Hardware Overhead Goal

Metric	Phase 1	Phase 2	Phase 3
Performance	<50%	<25%	<10%
Power	<10%	<10%	0%
Area	<100%	<50%	<30%
Security	3 CWEs	5 CWEs	7 CWEs
Scalability	1 CPU Simulation	3 CPUs	3 CPUs

SSITH Metrics

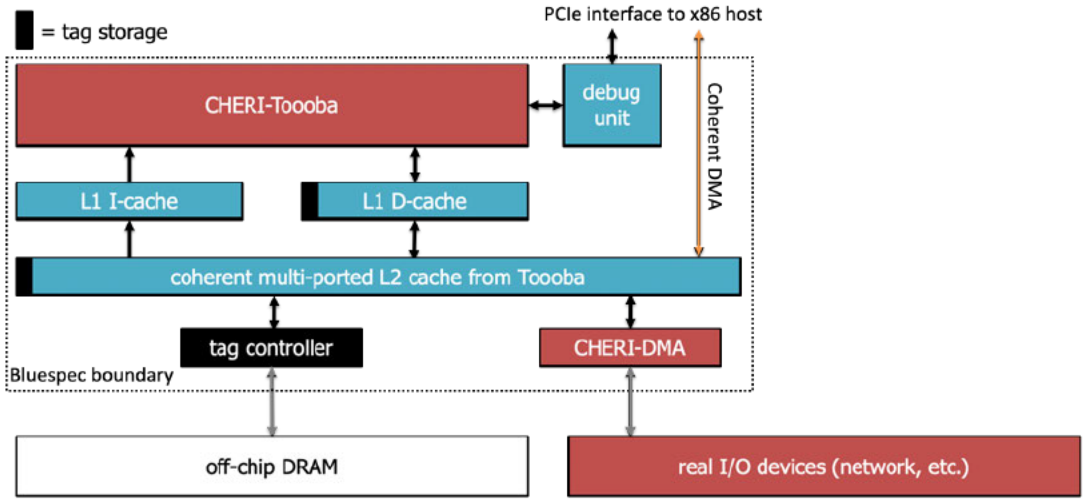
SSITH technology demonstrated in FPGA on 3 RISC-V CPUs

- P1 32-bit embedded processor (Piccolo/Rocket)
- P2 64-bit Linux processor (Flute/Rocket)
- P3 64-bit high-performance processor (Toooba/BOOM)

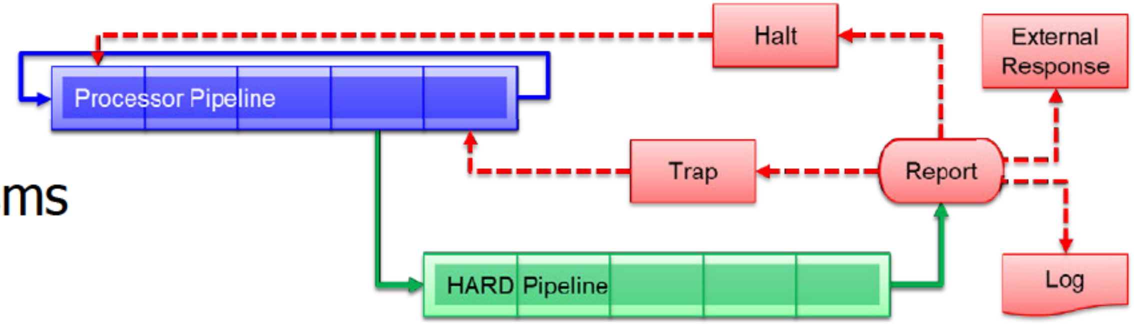
Hardware-based security can be implemented with low overhead

SSITH Phase 3 objectives

- SRI/Cambridge (TA-1)
  - Capabilities for software compartmentalization
  - CHERI-DMA protections
    - Protections in systems with untrusted hardware
    - Protections in systems with untrusted firmware
  - CHERI effects on speculative execution attacks

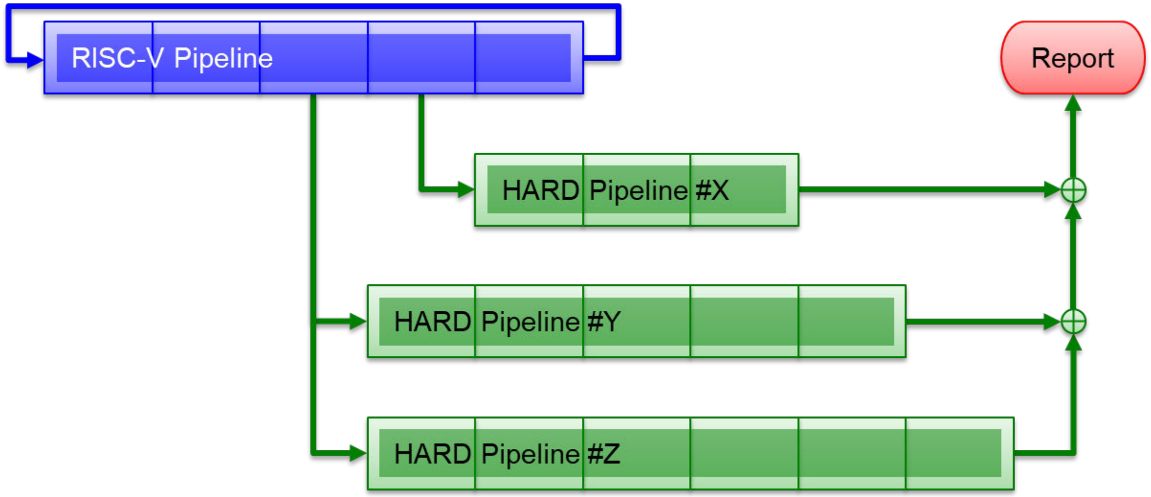


- Lockheed-Martin (TA-1)
  - Reduced compiler-specific behavior
  - Multi-process protections
  - Investigations of reporting/response mechanisms
  - Improved composability of pipelines



- Galois (TA-2)
  - Implementing tests for remaining CWE categories
  - Building virtualized test systems for transition partner evaluation

Lockheed Martin
HARD: Hardware Architecture Resilience by Design



Multiple Pipelines in Parallel

Technical Insight

- Multiple lightweight pipelines running in parallel leads to:
    - Lower PPA impacts
    - Customizable to different environmental needs
- Binary processing mitigates the need to recompile

Technical Approach

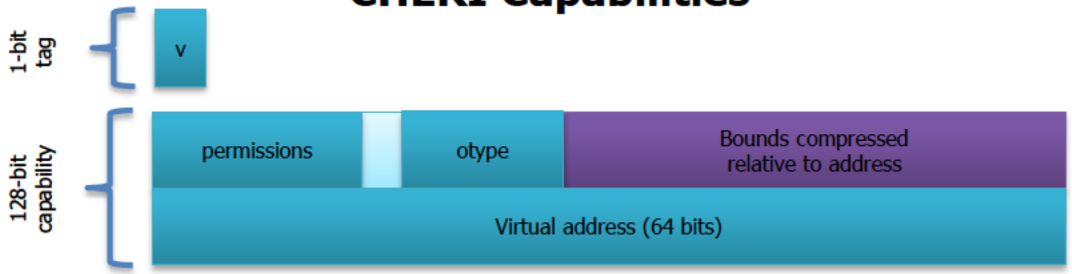
- HARD processor runs in parallel – low power overhead
- Multi-pronged: multiple lightweight hardware pipelines
- Each pipeline implements a specific security technique
- Binary executable code analysis and transformation (reverse engineering of binaries)
- Combination of tagging, fenced region, protection domains, per-thread keying, and memory encryption

HARD Features

- Minimally Invasive
    - Source code and toolchain for main pipeline is unchanged
    - No alteration of the main processor pipeline, just exposing select register and wires for monitoring
- Out of Band
    - Performance of main pipeline is unchanged
- Modular
    - HARD pipelines may be attached to other processor pipelines
    - Multiple HARD pipelines may be attached/combined to main pipeline to meet requirements of the application/environment

SRI International
ECATS: Extending the CHERI Architecture for Trustworthiness in SSITH

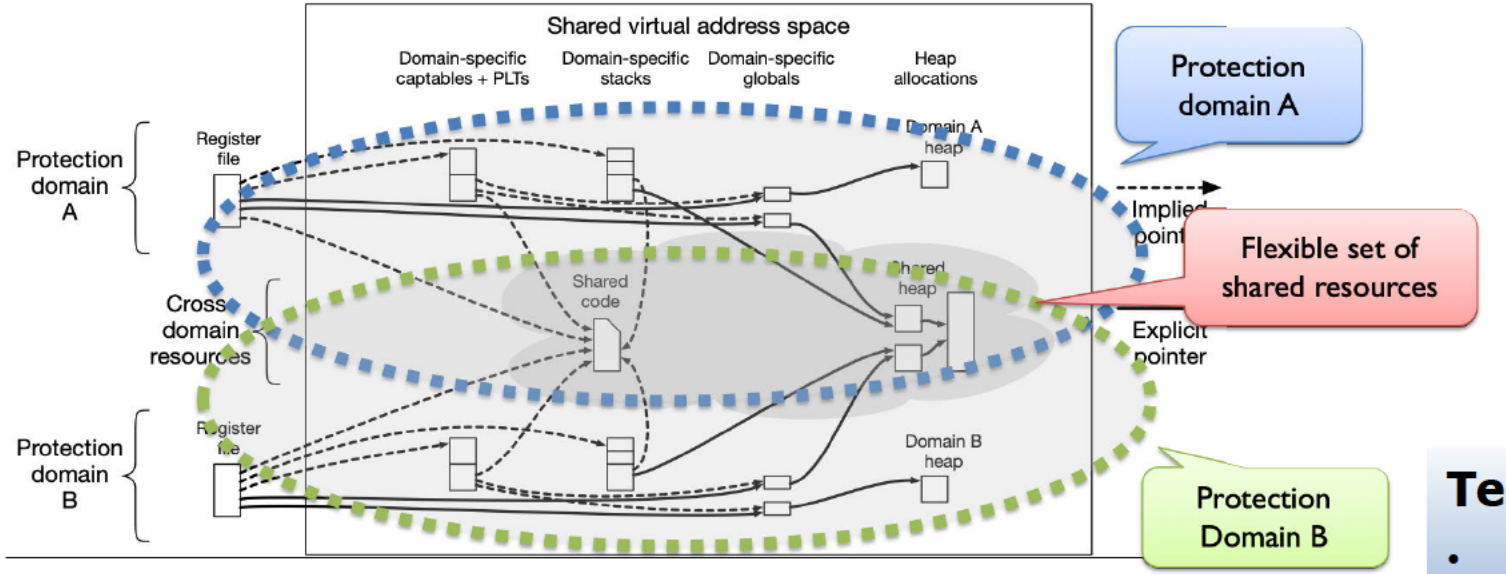
CHERI Capabilities


1-bit tag
v
128-bit capability
permissions
otype
Bounds compressed relative to address
Virtual address (64 bits)

CHERI capabilities extend pointers with a 1-bit tag as well as encoding (decreasing) permissions, and bounds.

Technical Approach

- Use *capabilities* for tagging and compartmentalization
- Apply principles of “least privilege” and “intentional use” to architecture and peripherals
  - Least privilege: limit access rights to the bare minimum
  - Intentional use: exercise of privileges must be explicitly selected
- Assume I/O devices cannot be fully trusted
- Instruction set is extended with capability instructions to
  - Query and manipulate capability fields
  - Use capabilities for load, store, jump targets, etc.



CHERI Features

- *Capabilities*
  - Implement strong memory protection
  - Implement scalable software compartmentalization
  - Extend to Accelerators, Direct Memory Access engines, microcontrollers, etc.
  - Protect I/O Memory Management Unit (IOMMU) Attacks

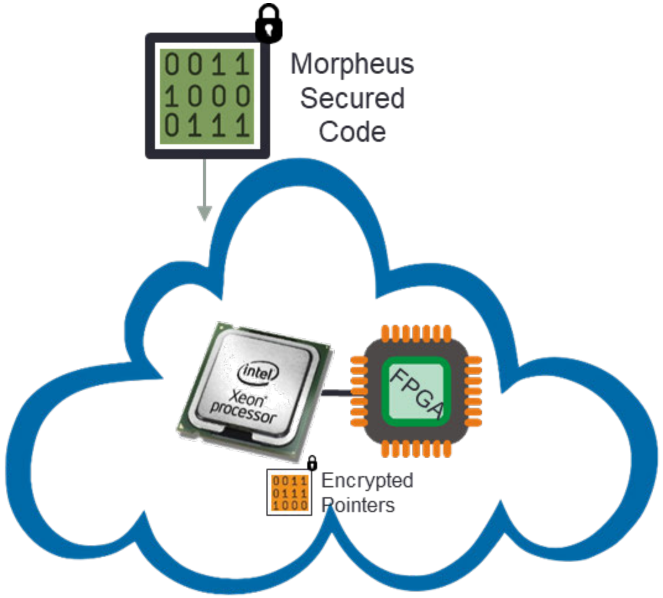
Technical Insight

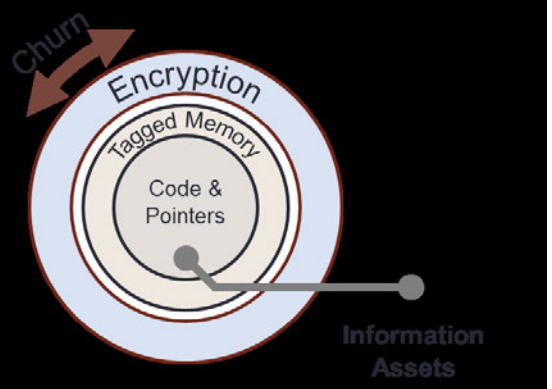
- CHERI *capabilities* are powerful enough to enforce memory access, compartmentalization, and I/O controls

Agita Labs
Morpheus Cloud: Protecting Critical Assets with Encryption

Morpheus Cloud Innovations

- Critical program assets are randomized with encryption
  - Code, code pointers, data pointers
  - Each domain has its own crypto keys
  - Decrypted at fetch, jumps and load/stores
- Assets remain encrypted in registers, memory, buses, I/O
  - Requires strong ciphers in the pipeline
- Churn re-encrypts a domain under a new random key
  - Places a time limit on penetrating encryption, determined in strength





Morpheus Cloud Technical Approach

- Run bulk of enterprise software on Xeon core
- Pointer processing occurs on always-encrypted data in the FPGA processor
- Cipher keys and pointer plaintext are sequestered to the FPGA, cannot be accessed by software
- Churn pointers using authenticated encryption

Technical Insight

Instead of fixing vulnerabilities, pervasively employ always-encrypted pointer protections to hide the critical information needed to attack

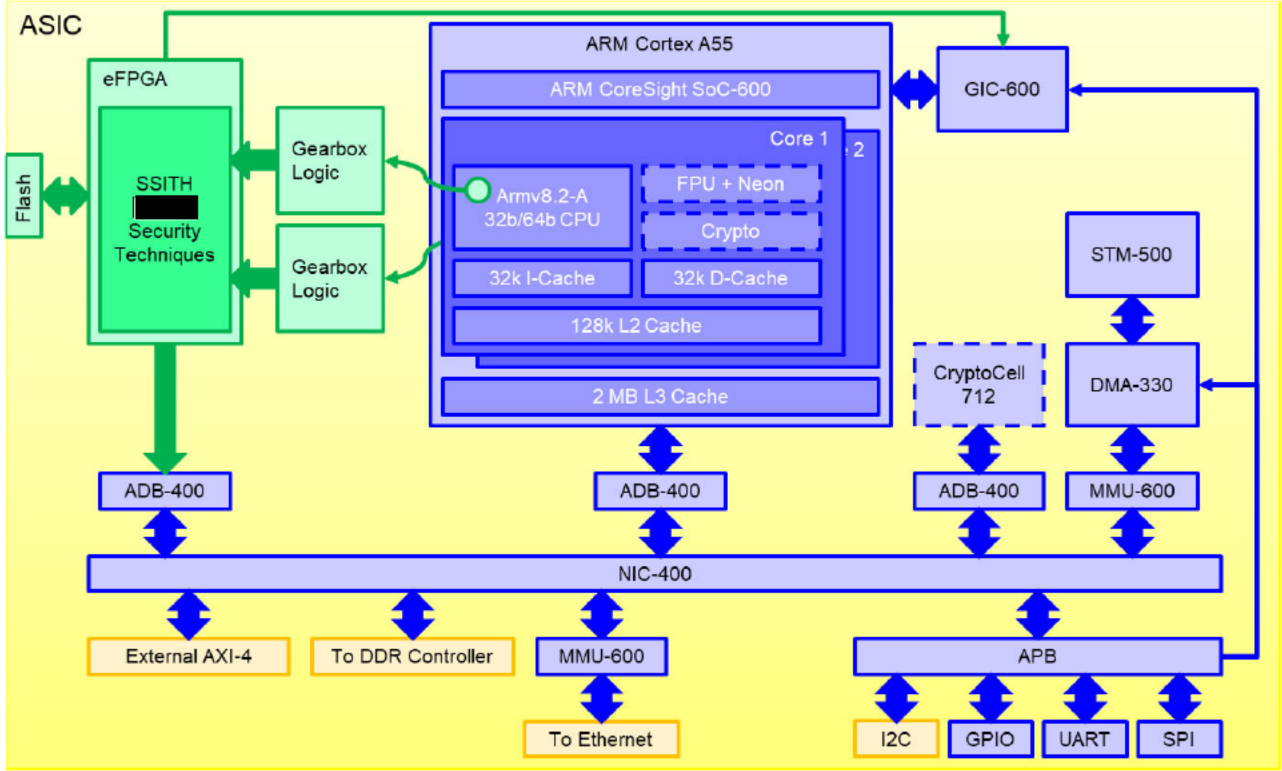
TA3: SSITH ARM ASIC

- Embedded FPGA
  - One-way monitoring and reporting of processor cores
  - Enables use of updated HARD pipelines
- Processor
  - Dual core A55
  - > 1 GHz
  - Cache: (32k L1, 128k L2, 2 MB L3)
  - Optional
    - FPU+Neon extension
    - Crytpo extension
    - CryptoCell 712
- Interfaces
  - DDR4: 64 bit with ECC
  - 10/100/1000 Ethernet
  - GPIO (24)
  - SPI, Quad SPI
  - External AXI-4 bus
  - I2C (2)
  - UART (4)
  - JTAG Debug port
    - CoreSight
    - STM-500

Tape Out Notes

- 12 nm at Global Foundries
- MPW shuttle run
- Assumption
  - October-November 2021 Tape-Out date
  - 5-6 months between tape-out and return of diced chips

ASIC (high-level)



BESSPIN: Balancing Evaluation of System Security Properties with Industrial Needs

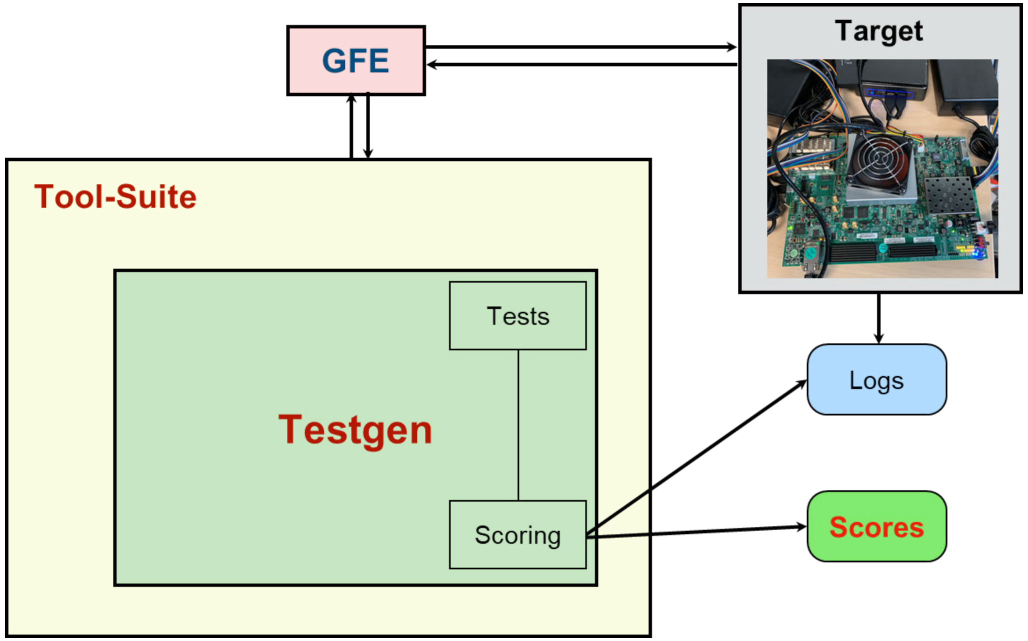
Testgen Security Evaluation Platform

- Tests weaknesses (CWEs) and not exploits (CVEs)
- Highly customizable to various TA-1 configurations
- Evolves more based on TA-1 teams feedback and processors behaviors
- Tool Suite generates PPAS metrics for TA-1 designs

BESSPIN Tool Suite

The BESSPIN Tool Suite includes tools to:

- Analyze, extract, and visualize the system architecture of an SoC
- Identify configurable aspects of a design and derive the feature model of a SoC
- Specify the security properties of a SoC by configuring different feature models for the CWE classes
- Specify the features of a platform under evaluation, including CPU, SoC, operating system, and compiler



Government-Furnished Equipment (GFE)

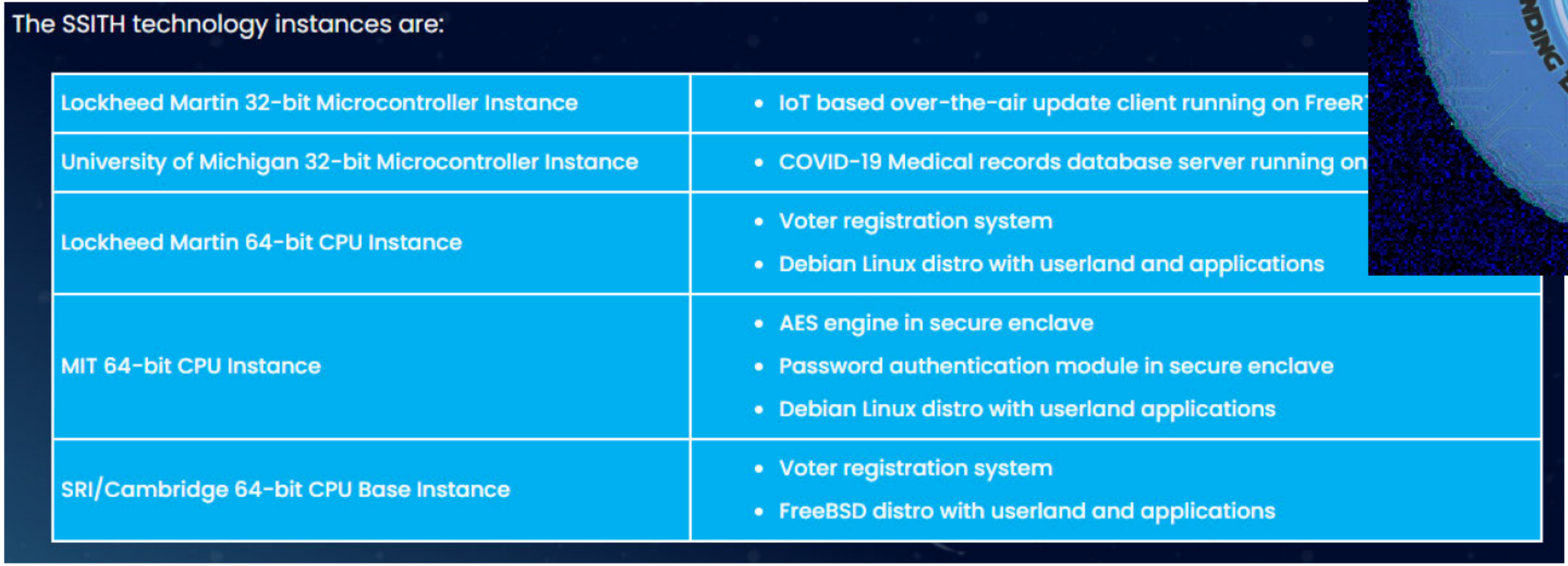
Galois developed three GFE RISC-V processors:

- P1: 32-bit embedded processor
- P2: 64-bit Linux processor
- P3: 64-bit high-performance (out-of-order) CPU

Chisel and BSV versions delivered to TA-1 performers

Bug Bounty Event: Finding Exploits to Thwart Tampering (FETT)

- **DARPA's first ever bug bounty program**
- Crowd-Sourced Red Team: Stress-test SSITH using real world cyber exploits executed by white hat hackers
- Validate SSITH's security benchmarking tool
- 07 – 09/2020: https://fett.darpa.mil



The SSITH technology instances are:

Lockheed Martin 32-bit Microcontroller Instance	• IoT based over-the-air update client running on FreeR
University of Michigan 32-bit Microcontroller Instance	• COVID-19 Medical records database server running on
Lockheed Martin 64-bit CPU Instance	• Voter registration system • Debian Linux distro with userland and applications
MIT 64-bit CPU Instance	• AES engine in secure enclave • Password authentication module in secure enclave • Debian Linux distro with userland applications
SRI/Cambridge 64-bit CPU Base Instance	• Voter registration system • FreeBSD distro with userland and applications

Tech Transition Strategy and Progress

Commercialization

- ARM is investigating SRI/Cambridge technology
  - The United Kingdom is funding the development of a CHERI-ARM ASIC
- SRI/Cambridge working with Microsoft and Google to improve OS security
- Dover Microsystems has licensed Draper Labs technology and designed ASICs with SSITH safeguards for NXP and Cadence/Tensilica
- Agita Labs startup out of University of Michigan

Influences or Establishes a Defined Technology Standard

- SRI/Cambridge formal semantics instructions set architecture (ISA) definition adopted by RISC-V community
- SSITH IOMMU security recommendations adopted in USB 4 standard
- MIT, UCSD, and SRI intend to publish software and IP via open source

Follow-on Development by a DoD Component

- Potential SSITH ASIC projects:
  - an Electronic Warfare RF SoC
  - an F-35 Cyber-network interface
  - Avionics Edge Computing
  - Power Plant Control

SSITH is working and transitioning

Major Achievements

Key accomplishments: Multiple demonstrations of secure architectures against 5 classes of hardware weaknesses

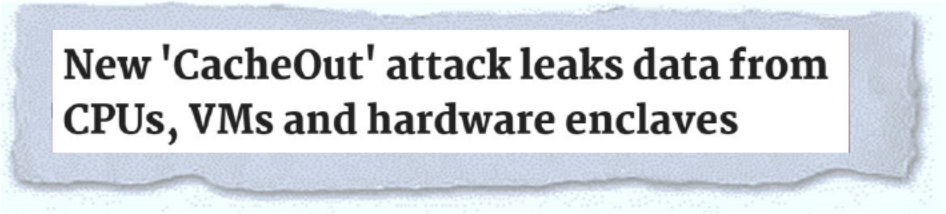
- All 3 classes of CPUs working on FPGA emulators
- Booted FreeRTOS and Linux operating systems
- Demonstrated information leakage mitigations against Spectre, MeltDown, and CacheOut attacks
- IOMMU Thunderbolt mitigations have been adopted by USB4 standard
- A toolbox of technologies have been developed which can trade:
  - Parallel pipelines for low performance impacts but larger area
  - Domain Specific Languages allow for updating security policies
  - Unmodified code runs with improved security
  - Recompiled code runs with maximum security

SSITH will:

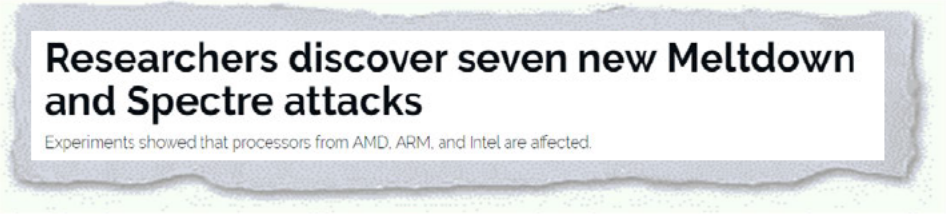
- Eliminate some hardware weaknesses and reduce the severity of others
- Allow legacy DoD applications to be protected without recompilation

SSITH will not:

- Have any impact on logic based software-based weaknesses
- Protect against unknown hardware flaws

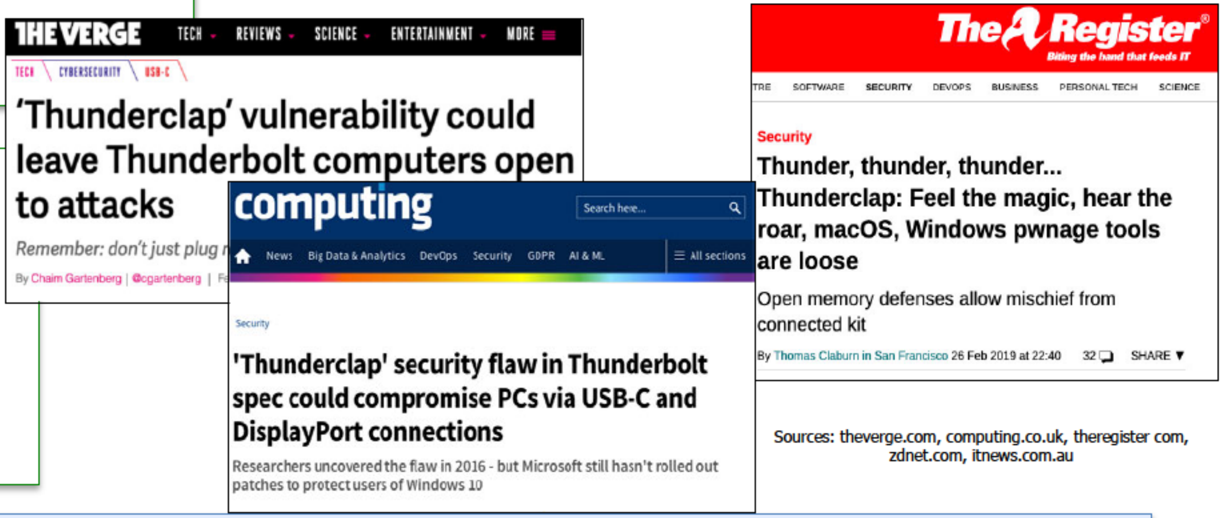


New 'CacheOut' attack leaks data from CPUs, VMs and hardware enclaves



Researchers discover seven new Meltdown and Spectre attacks

Experiments showed that processors from AMD, ARM, and Intel are affected.



‘Thunderclap’ vulnerability could leave Thunderbolt computers open to attacks

Remember: don’t just plug r

By Chaim Gartenberg | @cgartenberg | Fe

'Thunderclap' security flaw in Thunderbolt spec could compromise PCs via USB-C and DisplayPort connections

Researchers uncovered the flaw in 2016 - but Microsoft still hasn't rolled out patches to protect users of Windows 10

Thunder, thunder, thunder... Thunderclap: Feel the magic, hear the roar, macOS, Windows pwnage tools are loose

Open memory defenses allow mischief from connected kit

By Thomas Claburn in San Francisco 26 Feb 2019 at 22:40 32 SHARE ▼

Sources: theverge.com, computing.co.uk, theregister com, zdnet.com, itnews.com.au

SSITH protects against all classes of hardware weaknesses with minimal overhead

DISTRIBUTION A: Approved for Public Release, Distribution Unlimited

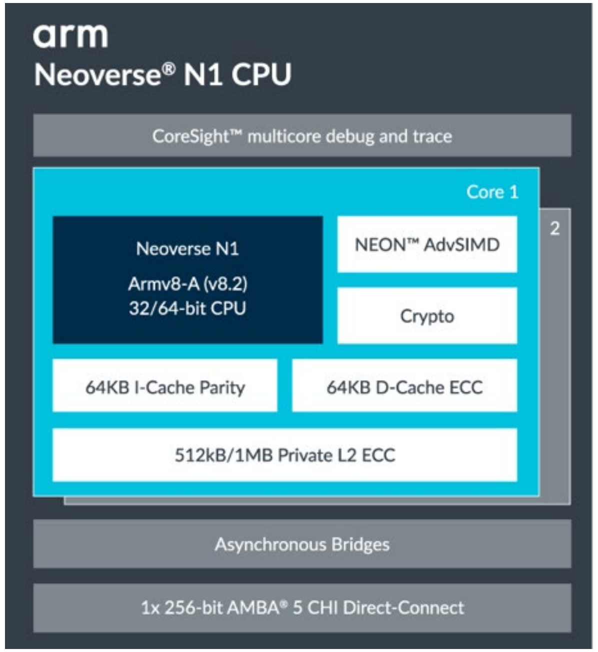
United Kingdom Research and Innovation Collaboration

- 5-year “Digital Security by Design” UK Research and Innovation (UKRI) program
- £185M funding
- University of Cambridge and ARM (SSITH performers and SRI subs)
- Develop an industrial demonstrator of a Capability architecture: the Morello Board
  - Superset of CHERI (Capability) architecture
  - Prototype technology that could be migrated to ARM architecture
  - Used for software experimentation and evaluation
  - Quad core high-end CPU based on Neoverse N1

Morello Cherry (CHERI)

Collaboration with UKRI

- Engaging in exploratory discussions with UK
  - Share technical progress
  - Access to prototype hardware
  - Avoid duplication of effort
- Drafting Project Arrangement document
- Planning joint review/kickoff meeting in the Spring



arm

Neoverse® N1 CPU

CoreSight™ multicore debug and trace

Core 1

2

Neoverse N1

Armv8-A (v8.2)
32/64-bit CPU

NEON™ AdvSIMD

Crypto

64KB I-Cache Parity

64KB D-Cache ECC

512kB/1MB Private L2 ECC

Asynchronous Bridges

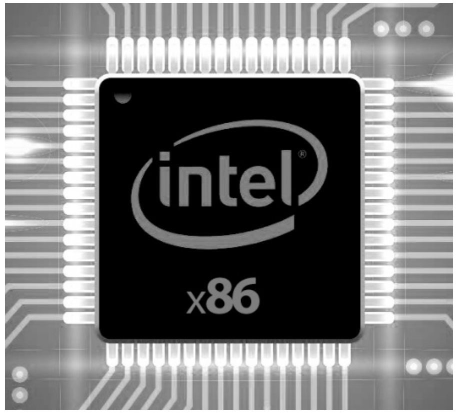
1x 256-bit AMBA® 5 CHI Direct-Connect

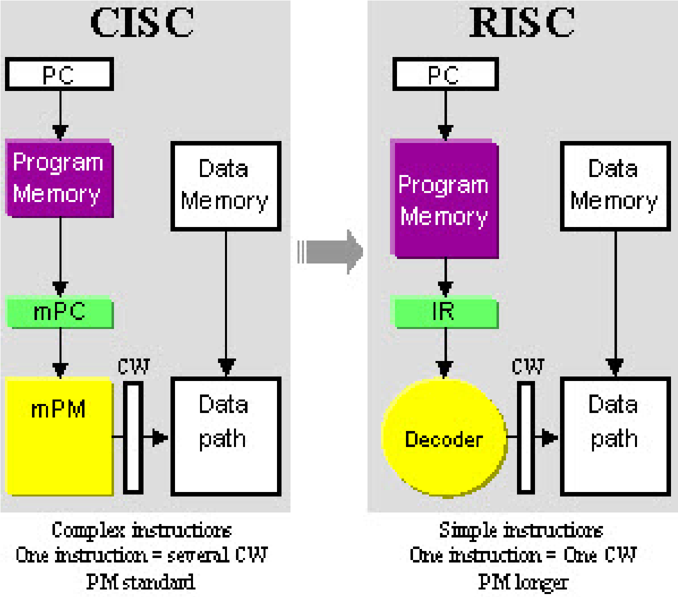
Extension to x86 Architectures

- ARM / RISC – predominate in low-power mobile and IoT
- x86 – Intel proprietary CISC (complex instruction set computing) – predominate in server / personal computing

Can we implement SSITH concepts in a CISC architecture?

- x86 is the most used ISA throughout DoD and commercial industry. Secure x86 is the number one transition partner request and paramount to national security
- DARPA is interested in the design and fabrication of secure x86 prototype chips which would incorporate SSITH like security features for DoD evaluation. If successful this effort would:
  - Enable rapid maturation, commercialization, transition, and fielding of SSITH secure technologies into x86 for modernization of DoD systems
  - Create U.S. based IP and a domestic commercial product line accessible to DoD
  - Provide DoD and the defense industrial base early access to advanced secure x86 processors
- Performed market research with all 3 x86 licensees to assess feasibility





https://wccftech.com/intel-developing-new-x86-uarch-succeed-core-generation/
https://www.watelectronics.com/what-is-risc-and-cisc-architecture/

SSITH Limitations

- SSITH addresses weaknesses that can be mitigated by hardware
  - Large numbers of vulnerabilities can be addressed: 70% of all cyber attacks are a result of buffer errors
  - But, vulnerabilities exist in higher-levels of the software stacks that SSITH does not address
    - Script injection
    - SQL injection
    - Weak custom authentication protocols

- It is challenging to defend against deliberately inserted weaknesses
  - Backdoors
  - Hard-coded credentials

- Full integration of SSITH security features with operating systems and common application software will be a prolonged transition activity

- There are never perfect solutions to security
  - Must expect the development of novel exploit techniques that violate SSITH assumptions

Summary

- The SSITH program will meet goals and expectations
  - All performers have demonstrated that their technical approaches will protect against hardware weaknesses
  - In Phase two of the program protections against 6 of 7 CWE classes were developed

- Strong transition strategy with multiple commercial and DoD partners
  - International cooperation efforts
  - Incorporation in industry standards
  - Potential buy-in by major companies, Google, Microsoft, …
  - SSITH ASIC to be tested in DoD-relevant applications

- Major impact going forward
  - Performers have developed multiple technologies to address problem
  - Can protect against attacks to legacy systems without redesign / recompilation
  - Industry is investigating the incorporation of SSITH technologies into COTS products
  - Possible extension to x86 architectures

Secure hardware can be implemented with minimal overhead