JOSEPH KINIRY

PRINCIPAL SCIENTIST, GALOIS
PRINCIPLED CEO & CHIEF SCIENTIST, FREE & FAIR

DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

SECURE SYSTEMS: VOTING

THIS RESEARCH WAS DEVELOPED WITH FUNDING FROM THE DEFENSE ADVANCED RESEARCH PROJECTS AGENCY (DARPA).

THE VIEWS AND CONCLUSIONS CONTAINED IN THIS DOCUMENT ARE THOSE OF THE AUTHORS AND SHOULD NOT BE INTERPRETED AS REPRESENTING THE OFFICIAL POLICIES, EITHER EXPRESSED OR IMPLIED, OF THE U.S. GOVERNMENT.

COMPUTER SECURITY IS WHACK-A-MOLE

- software has bugs
- **lots** of bugs
- commercial code has about
  1 bug for every 20 lines of code
- many of those bugs are innocuous—think a rendering glitch or a 1 in a billion failure
- some of those bugs are a bummer—think about the last document you lost in a crash
- some of those bugs are critical and are leveraged by malicious actors—think WannaCry

PHOTO BY FLICKR USER TPAPI RELEASED UNDER CC BY-NC-SA 2.0

HARDWARE: THE BUCK STOPS HERE

- hardware does **very** little to help
- dozens of generations of hardware were designed without security requirements
- good guys and bad guys discover hardware vulnerabilities every day
- bugs are fixed and flawed systems are redesigned

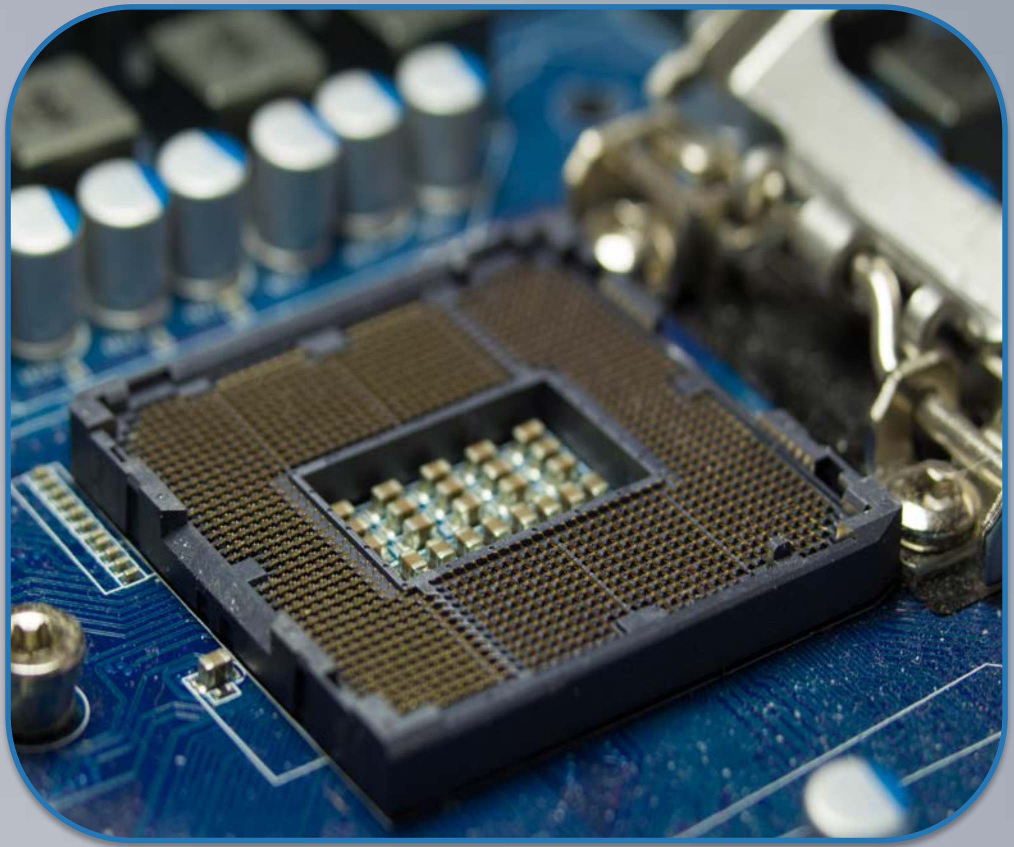


PHOTO RELEASED UNDER CC0 VIA PIXHERE.COM

DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

HARDWARE: THE BUCK STOPS HERE

- hardware does **very** little to help
- dozens of generations of hardware were designed without security requirements
- good guys and bad guys discover hardware vulnerabilities every day
- bugs are fixed and flawed systems are redesigned
- but we are living with a legacy of trillions of lines of code…

and therefore **billions** of bugs

THE DEVIL (OR DAEMON) IS IN DEVICES

- hardware devices—from DVDs to USBs to Verilog IP—are routinely created without an adversarial threat model…
- and devices **never** get patched
- so systems are compromised… **and it is not the user's fault**

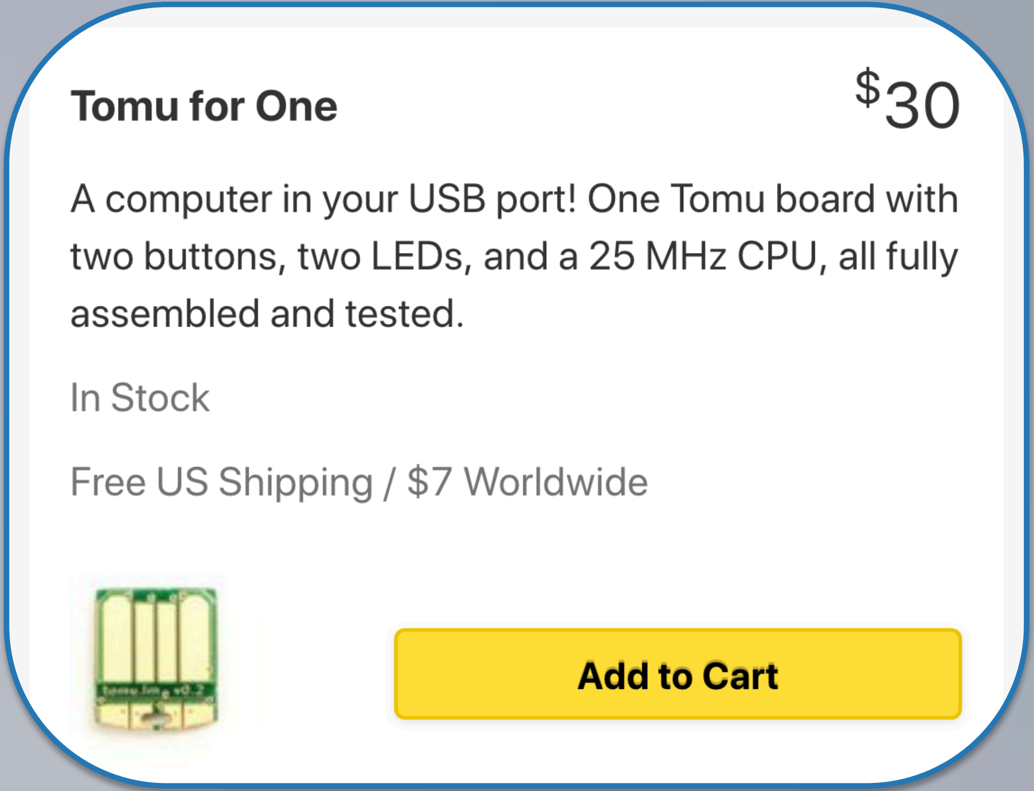
BSD DAEMON COPYRIGHT MARSHALL KIRK MCKUSICK
THIS IMAGE COMES FROM WALNUT CREEK CD-ROM FOR FREEBSD 2.0
ORIGINAL ARTWORK BY JOHN LASSETER

DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

THE DEVIL (OR DAEMON) IS IN DEVICES

- hardware devices—from DVDs to USBs to Verilog IP—are routinely created without an adversarial threat model…
- and devices **never** get patched
- so systems are compromised… **and it is not the users fault**

- the consumer market for spy devices—technologies once only within the capabilities of advanced nation states are *now for sale, for cheap, and are open hardware*



Tomu for One $30

A computer in your USB port! One Tomu board with two buttons, two LEDs, and a 25 MHz CPU, all fully assembled and tested.

In Stock

Free US Shipping / $7 Worldwide

CROWDSUPPLY.COM SHOPPING CART SCREENSHOT

WHAT HAPPENS WHEN ADVERSARIES DECIDE TO REALLY MAKE A POINT?

WE MUST CREATE A NEW GENERATION OF SECURE HARDWARE

DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

SYSTEM DESIGN & HARDWARE SECURITY: A DEPRESSING ANALOGY

- every piece of silicon is a microscopic, concurrent, distributed system

- dozens to hundreds of engineers
- hundreds of person years
- millions of lines of RTL
- billions of transistors

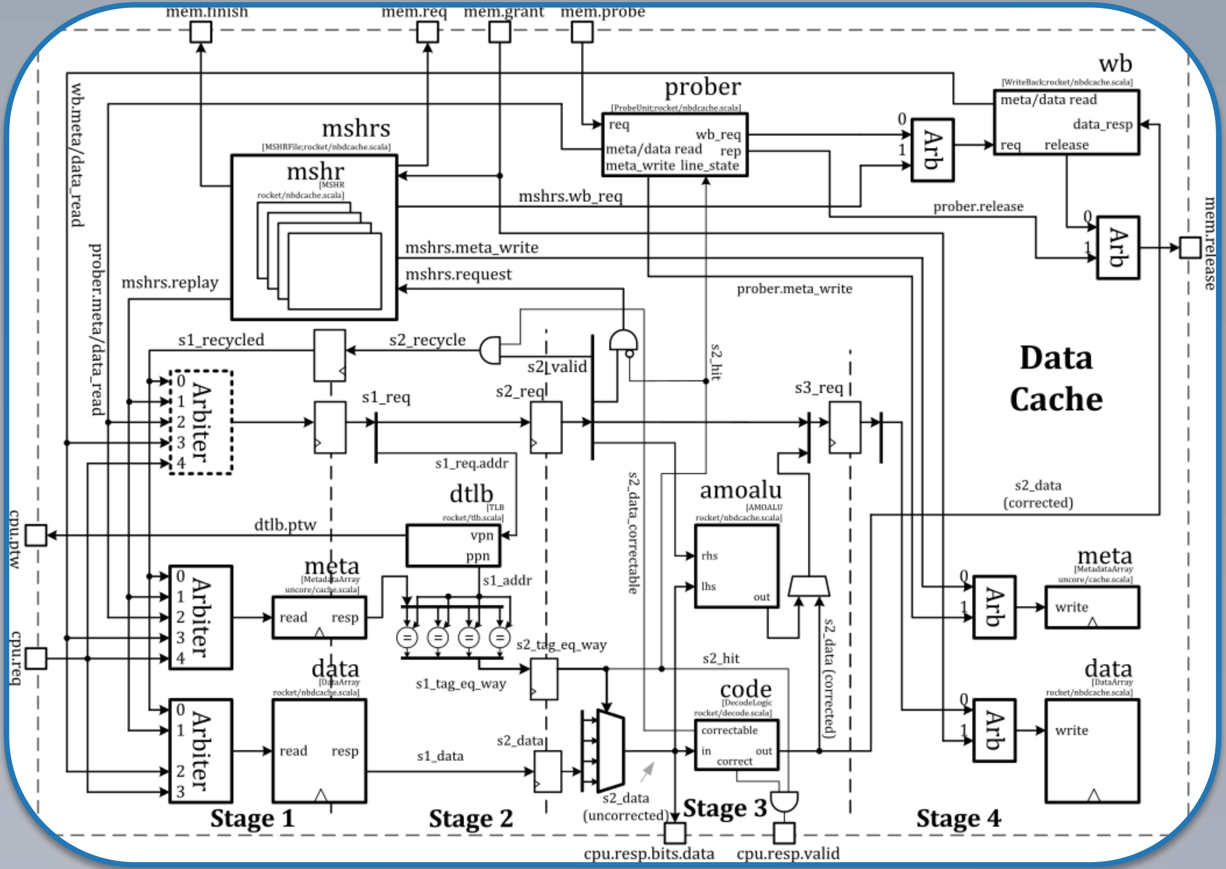


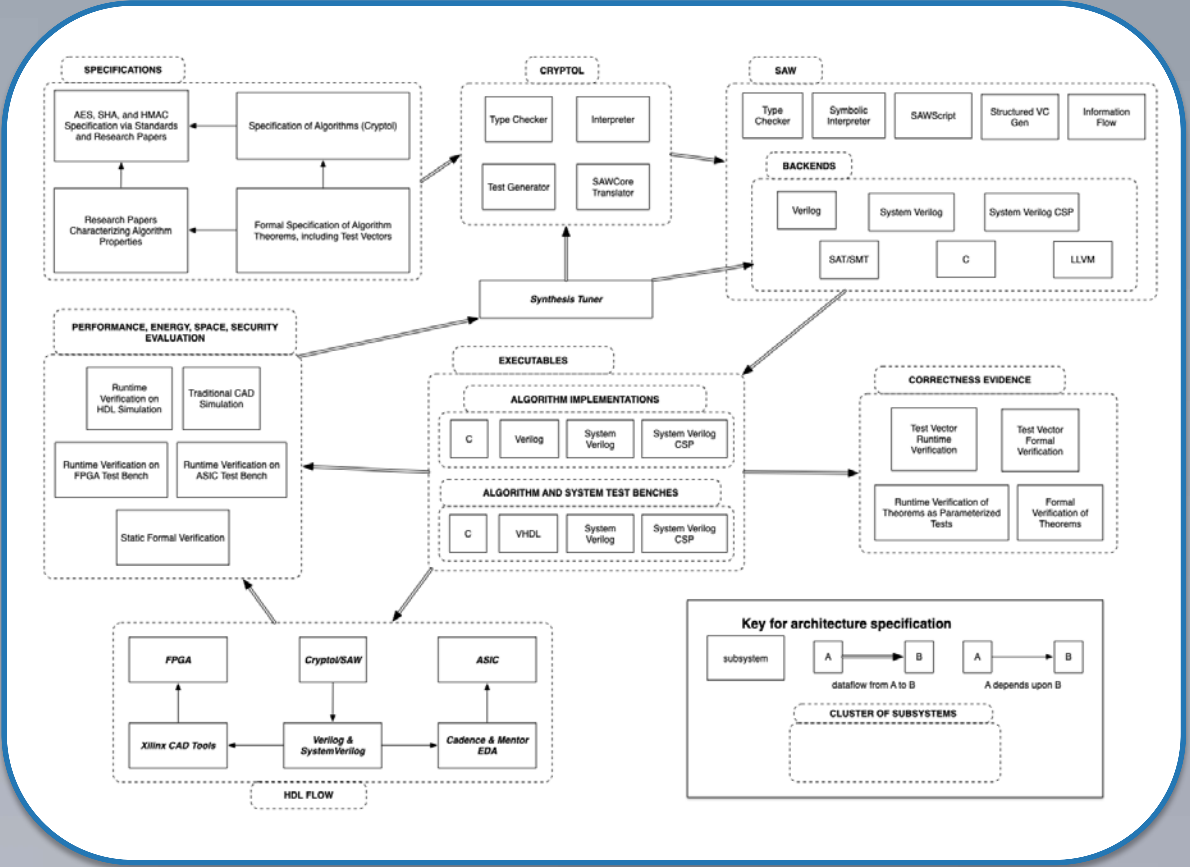
ILLUSTRATION FROM LOWRISC RELEASED UNDER CC BY-SA 4.0

DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

SYSTEM DESIGN & HARDWARE SECURITY: A DEPRESSING ANALOGY

- most modern software systems are human-scale, concurrent, distributed systems

- tens to hundreds of developers
- hundreds of person years
- millions of lines of source code
- billions of lines of object code



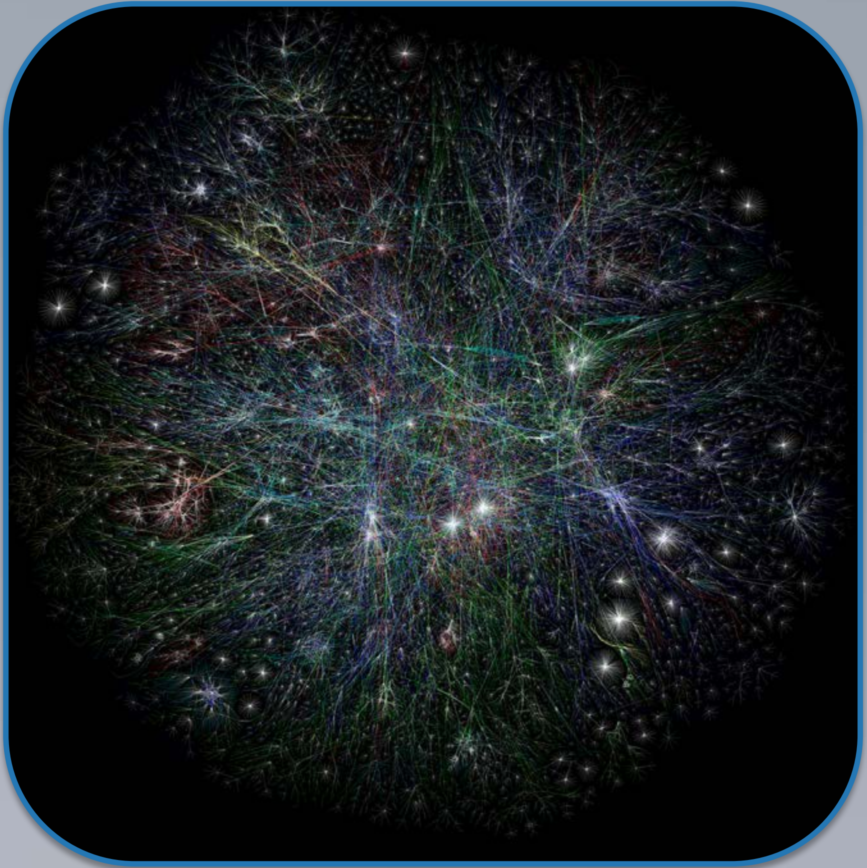
SPECIFICATIONS
AES, SHA, and HMAC Specification via Standards and Research Papers
Specification of Algorithms (Cryptol)
Research Papers Characterizing Algorithm Properties
Formal Specification of Algorithm Theorems, including Test Vectors
CRYPTOL
Type Checker
Interpreter
Test Generator
SAWCore Translator
SAW
Type Checker
Symbolic Interpreter
SAWScript
Structured VC Gen
Information Flow
BACKENDS
Verilog
System Verilog
System Verilog CSP
SAT/SMT
C
LLVM
Synthesis Tuner
PERFORMANCE, ENERGY, SPACE, SECURITY EVALUATION
Runtime Verification on HDL Simulation
Traditional CAD Simulation
Runtime Verification on FPGA Test Bench
Runtime Verification on ASIC Test Bench
Static Formal Verification
EXECUTABLES
ALGORITHM IMPLEMENTATIONS
C
Verilog
System Verilog
System Verilog CSP
ALGORITHM AND SYSTEM TEST BENCHES
C
VHDL
System Verilog
System Verilog CSP
CORRECTNESS EVIDENCE
Test Vector Runtime Verification
Test Vector Formal Verification
Runtime Verification of Theorems as Parameterized Tests
Formal Verification of Theorems
FPGA
Cryptol/SAW
ASIC
Xilinx CAD Tools
Verilog & SystemVerilog
Cadence & Mentor EDA
HDL FLOW
Key for architecture specification
subsystem
A
B
dataflow from A to B
A
B
A depends upon B
CLUSTER OF SUBSYSTEMS

SYSTEM DESIGN & HARDWARE SECURITY: A DEPRESSING ANALOGY

- most modern computation is a global-scale, concurrent, distributed system…

  the Internet

- security is prolific and baked in
- systems never trust their peers
- constant evolution of security technologies to mitigate threats

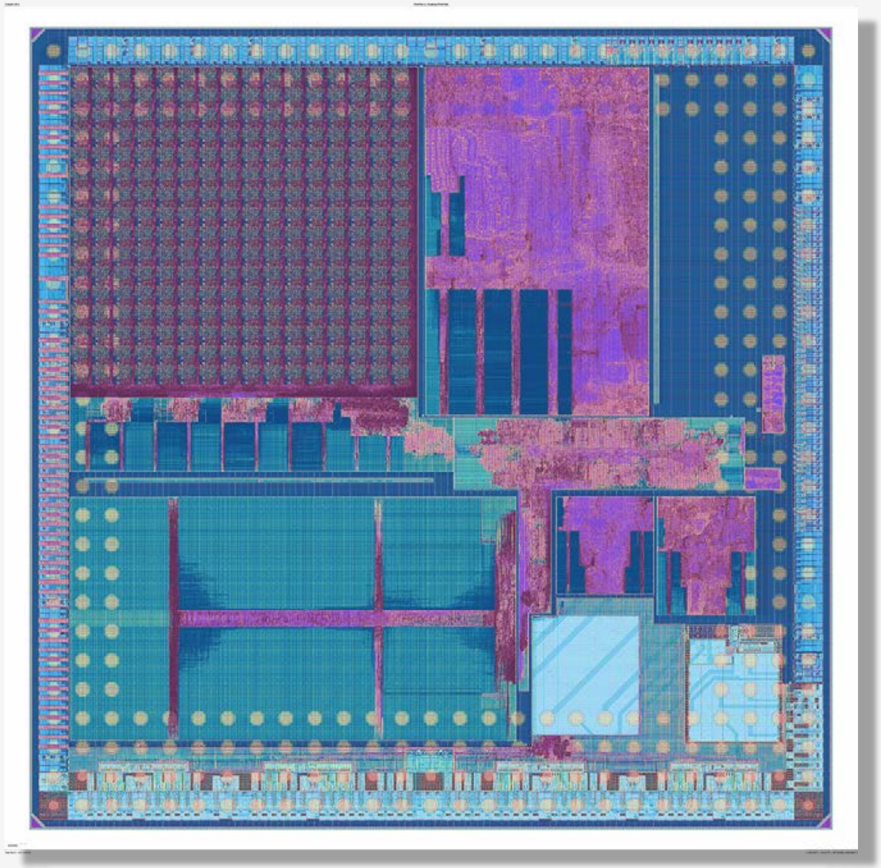


RENDER FROM WIKIPEDIA COMMONS RELEASED UNDER CC BY 2.5

SYSTEM DESIGN & HARDWARE SECURITY: A DEPRESSING ANALOGY

- now imagine the Internet…
- **without any security**…
- where nothing can be patched…
- only a handful of vendors create every subsystem…
- and no one want to pay it forward for security…

that is modern hardware.



DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

A MATTER OF PRIORITIES

- companies must prioritize what their customers demand
- historically that means…
  - better **P**erformance
  - lower **P**ower
  - smaller **A**rea
- our priorities—defense, businesses, and consumers alike—are changing…
- now, and into the future, we must balance **PPA** against **Security…**
  thus **PPAS** (PPAS is SWaP for ASICs)

RENDER FROM CCPIXS.COM RELEASED UNDER CC BY 3.0

DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

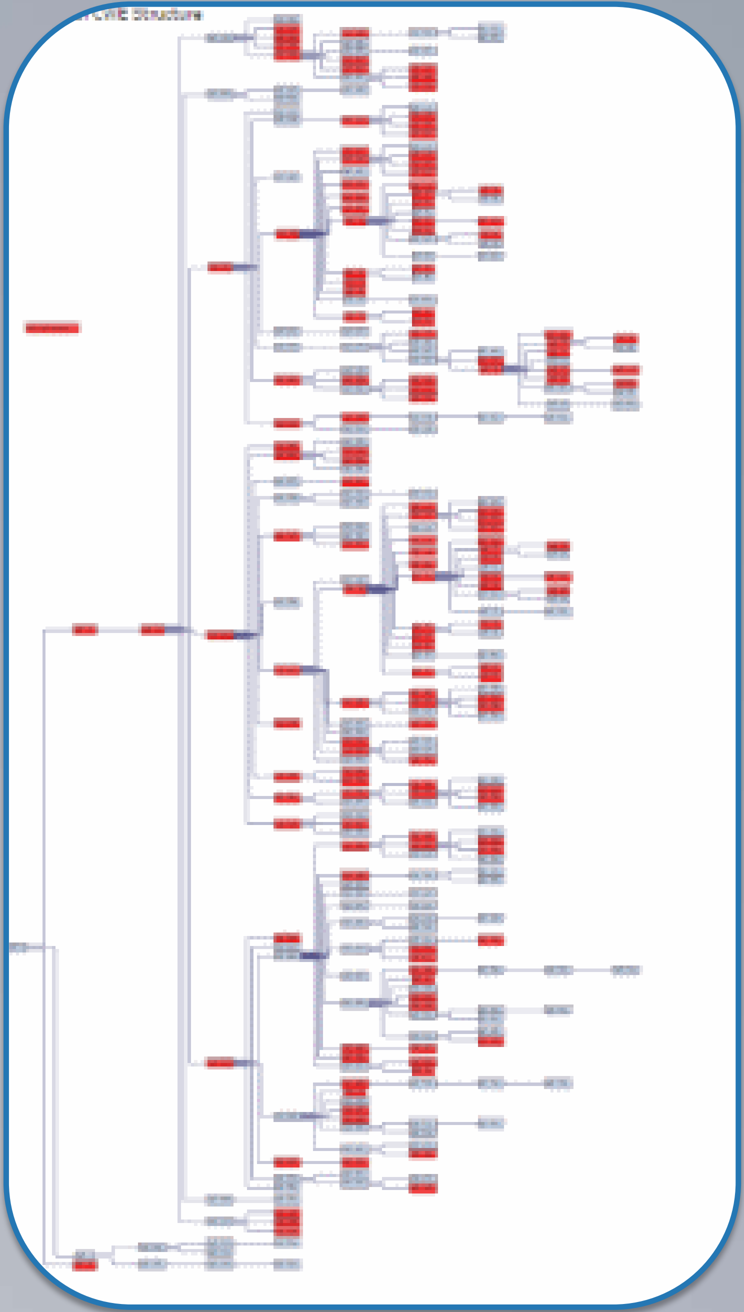
SSITH IN A NUTSHELL

- **secure hardware FTW!**
- goal is to eliminate most classes of software vulnerabilities
- open source, soft-core RISC-V on FPGAs as the demo platform
- six teams developing 18 SoCs
- each team augments three baseline RISC-V SoCs to make them secure
  - a 32 bit microcontroller and
    two 64 bit CPUs (one OOO)
- security approaches are all over the map, including tagging, enclaves, novel crypto, and AI

DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

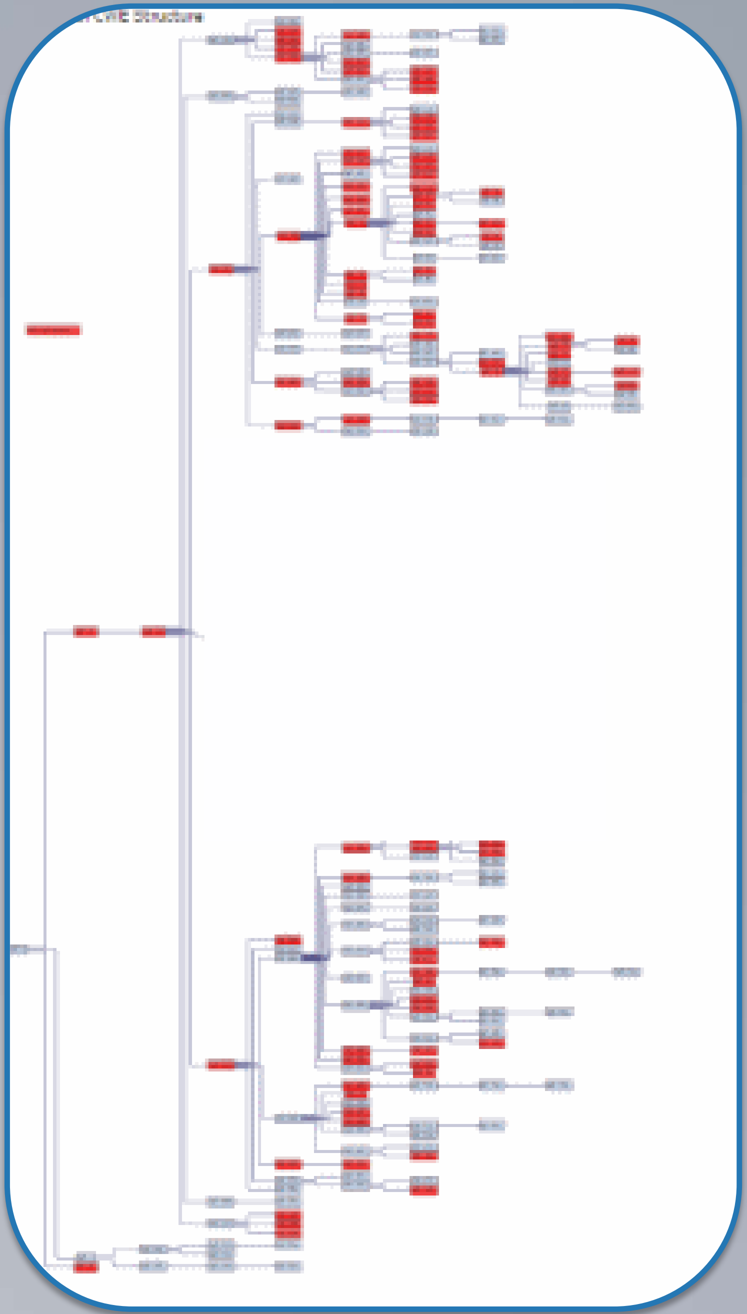
MITIGATING SOFTWARE VULNERABILITIES WITH HARDWARE

- SSITH CPUs must be backwards compatible & run existing binaries
- these binaries have vast numbers of exploitable vulnerabilities
- software vulnerabilities are classified using NIST CWE classes, which form a subtyping tree depicted at right



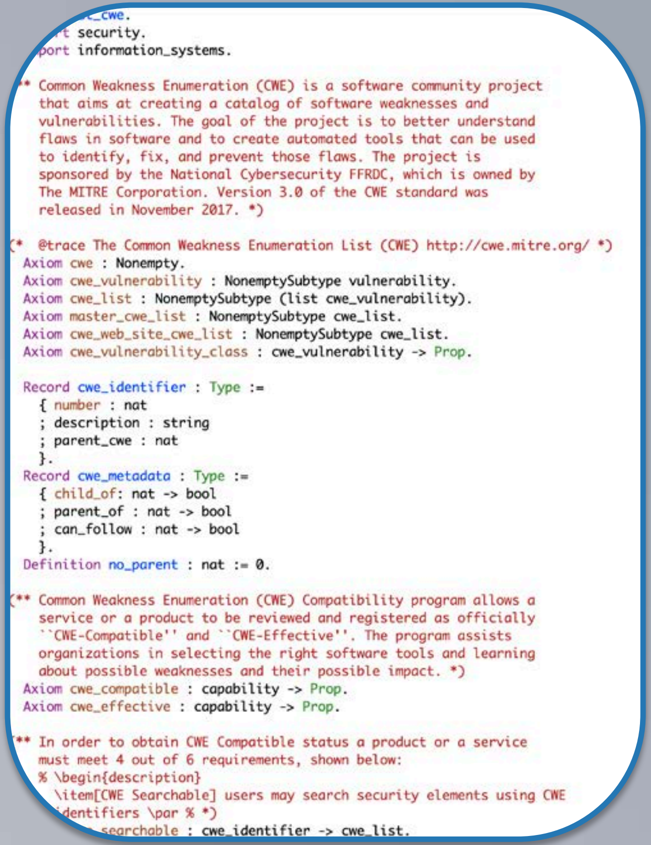
MITIGATING SOFTWARE VULNERABILITIES WITH HARDWARE

- SSITH CPUs must be backwards compatible & run existing binaries
- these binaries have vast numbers of exploitable vulnerabilities
- software vulnerabilities are classified using NIST CWE classes, which form a subtyping tree depicted at right
- SSITH CPUs mitigate specific CWE types, thus pruning subtrees of software vulnerabilities away



GALOIS'S ROLE IN SSITH: THE BESSPIN PROJECT

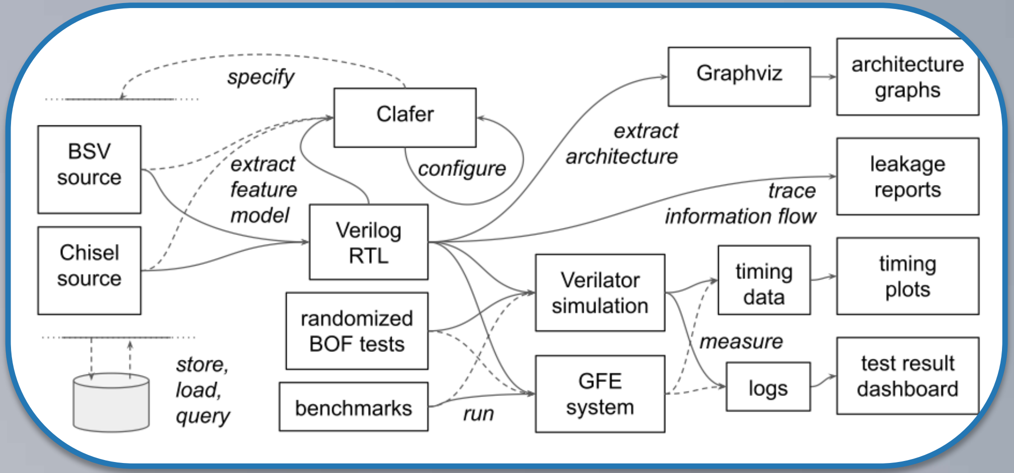
- make precise the seven classes of vulnerabilities in the program
  - what is a memory error, information leakage, etc.



```
    _cwe.
  t security.
port information_systems.

** Common Weakness Enumeration (CWE) is a software community project
   that aims at creating a catalog of software weaknesses and
   vulnerabilities. The goal of the project is to better understand
   flaws in software and to create automated tools that can be used
   to identify, fix, and prevent those flaws. The project is
   sponsored by the National Cybersecurity FFRDC, which is owned by
   The MITRE Corporation. Version 3.0 of the CWE standard was
   released in November 2017. *)

(*  @trace The Common Weakness Enumeration List (CWE) http://cwe.mitre.org/ *)
 Axiom cwe : Nonempty.
 Axiom cwe_vulnerability : NonemptySubtype vulnerability.
 Axiom cwe_list : NonemptySubtype (list cwe_vulnerability).
 Axiom master_cwe_list : NonemptySubtype cwe_list.
 Axiom cwe_web_site_cwe_list : NonemptySubtype cwe_list.
 Axiom cwe_vulnerability_class : cwe_vulnerability -> Prop.

 Record cwe_identifier : Type :=
   { number : nat
   ; description : string
   ; parent_cwe : nat
   }.
 Record cwe_metadata : Type :=
   { child_of: nat -> bool
   ; parent_of : nat -> bool
   ; can_follow : nat -> bool
   }.
 Definition no_parent : nat := 0.

(** Common Weakness Enumeration (CWE) Compatibility program allows a
   service or a product to be reviewed and registered as officially
   ``CWE-Compatible'' and ``CWE-Effective''. The program assists
   organizations in selecting the right software tools and learning
   about possible weaknesses and their possible impact. *)
 Axiom cwe_compatible : capability -> Prop.
 Axiom cwe_effective : capability -> Prop.

** In order to obtain CWE Compatible status a product or a service
   must meet 4 out of 6 requirements, shown below:
   % \begin{description}
     \item[CWE Searchable] users may search security elements using CWE
    identifiers \par % *)
     searchable : cwe_identifier -> cwe_list.
```

GALOIS'S ROLE IN SSITH: THE BESSPIN PROJECT

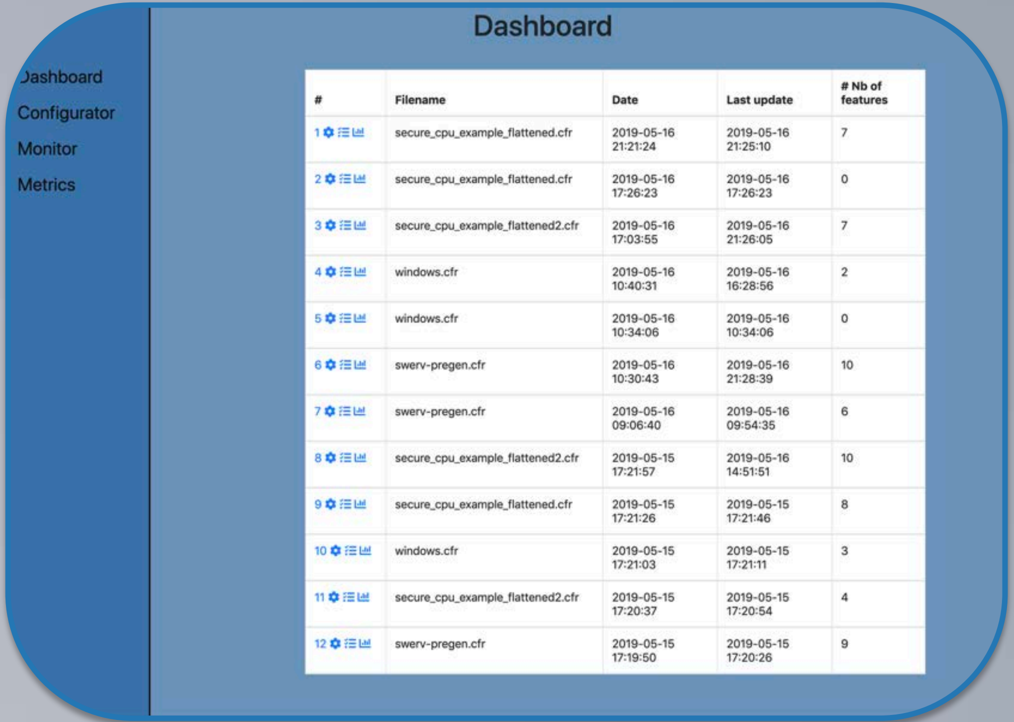
- make precise the seven classes of vulnerabilities in the program
- provide tools that integrate with normal design flows and generate evidence that a RISC-V SoC is **correct and secure**



galois

GALOIS’S ROLE IN SSITH: THE BESSPIN PROJECT

- make precise the seven classes of vulnerabilities in the program
- provide tools that integrate with normal design flows and generate evidence that a RISC-V SoC is **correct and secure**
- **create new EDA tools** that permit hardware designers to **objectively measure PPAS** and **optimize a SoC design for their particular requirements**
  - *SWAP/PPAS tradeoffs are evidence-based and transparent*



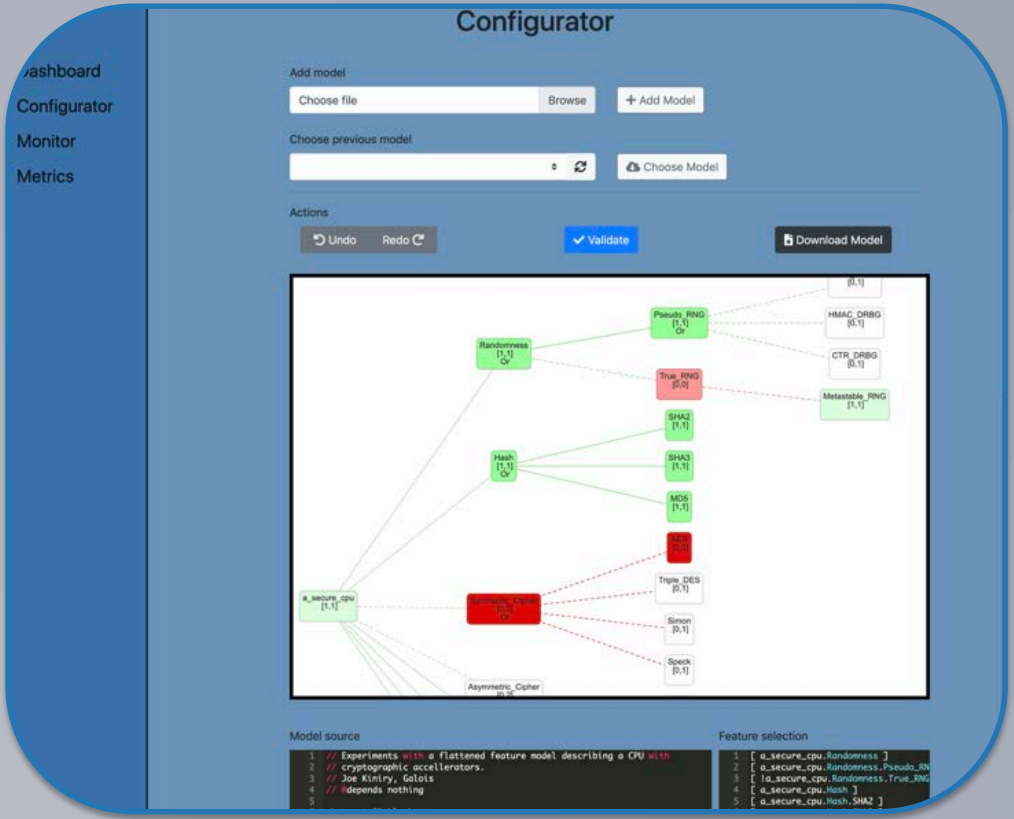
Dashboard

#	Filename	Date	Last update	# Nb of features
1	secure_cpu_example_flattened.cfr	2019-05-16 21:21:24	2019-05-16 21:25:10	7
2	secure_cpu_example_flattened.cfr	2019-05-16 17:26:23	2019-05-16 17:26:23	0
3	secure_cpu_example_flattened2.cfr	2019-05-16 17:03:55	2019-05-16 21:26:05	7
4	windows.cfr	2019-05-16 10:40:31	2019-05-16 16:28:56	2
5	windows.cfr	2019-05-16 10:34:06	2019-05-16 10:34:06	0
6	swerv-pregen.cfr	2019-05-16 10:30:43	2019-05-16 21:28:39	10
7	swerv-pregen.cfr	2019-05-16 09:06:40	2019-05-16 09:54:35	6
8	secure_cpu_example_flattened2.cfr	2019-05-15 17:21:57	2019-05-16 14:51:51	10
9	secure_cpu_example_flattened.cfr	2019-05-15 17:21:26	2019-05-15 17:21:46	8
10	windows.cfr	2019-05-15 17:21:03	2019-05-15 17:21:11	3
11	secure_cpu_example_flattened2.cfr	2019-05-15 17:20:37	2019-05-15 17:20:54	4
12	swerv-pregen.cfr	2019-05-15 17:19:50	2019-05-15 17:20:26	9

DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

GALOIS’S ROLE IN SSITH: THE BESSPIN PROJECT

- make precise the seven classes of vulnerabilities in the program
- provide tools that integrate with normal design flows and generate evidence that a RISC-V SoC is **correct and secure**
- **create new EDA tools** that permit hardware designers to **objectively measure PPAS** and **optimize a SoC design for their particular requirements**
  - *SWAP/PPAS tradeoffs are evidence-based and transparent*



```
// Experiments with a flattened feature model describing a CPU with
// cryptographic accellerators.
// Joe Kiniry, Galois
// @depends nothing

```

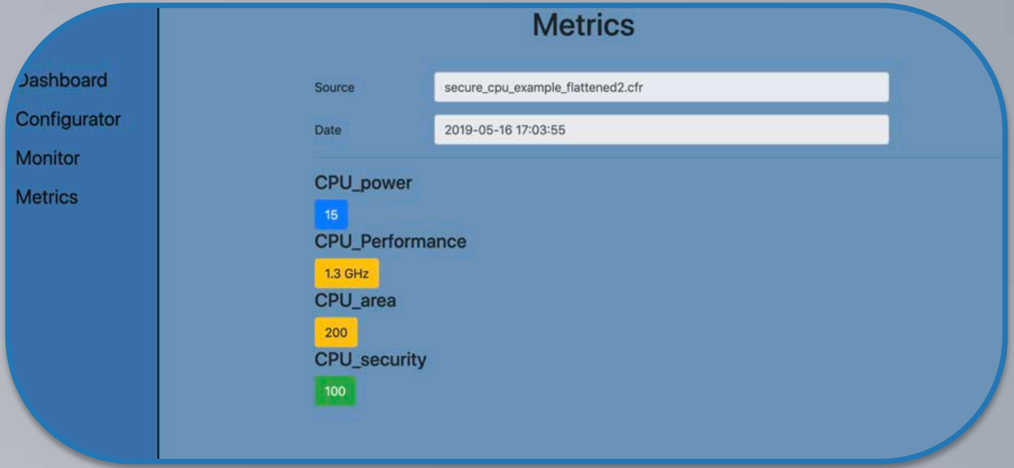
```
[ a_secure_cpu.Randomness ]
[ a_secure_cpu.Randomness.Pseudo_RN
[ !a_secure_cpu.Randomness.True_RNG
[ a_secure_cpu.Hash ]
[ a_secure_cpu.Hash.SHA2 ]
```

galois

DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

GALOIS'S ROLE IN SSITH: THE BESSPIN PROJECT

- make precise the seven classes of vulnerabilities in the program
- provide tools that integrate with normal design flows and generate evidence that a RISC-V SoC is **correct and secure**
- **create new EDA tools** that permit hardware designers to **objectively measure PPAS** and **optimize a SoC design for their particular requirements**
    - *SWAP/PPAS tradeoffs are evidence-based and transparent*



Metrics

Source: secure_cpu_example_flattened2.cfr

Date: 2019-05-16 17:03:55

CPU_power: 15

CPU_Performance: 1.3 GHz

CPU_area: 200

CPU_security: 100

galois

RED TEAMING SSITH SECURITY: AMBITIONS

- but how do we evaluate the security of a SoC?
- how can we possibly evaluate the security of six teams' work including 18 SoCs across six architectures, three OSs, and six compilers?
- a set of red teams to accomplish this goal would take years and cost seven figures and probably result in proprietary "we tried our best" reports
- these systems are meant to be secure…
  - **even when their entire design and implementation is public…**
  - **even when the system's network is compromised…**
  - **even when the adversary has a beachhead and can install malware!**

RED TEAMING SSITH SECURITY: EXECUTION

- why not **open up the red teaming to the world?**
- **go beyond** public exercises like the DARPA Urban Challenge and the CGC

- DARPA has decided to kick-off the entire security evaluation exercise in August in Las Vegas at DEF CON 2019

- we will also make available a low-cost hardware platform and pedagogical materials to facilitate teaching and learning, course and research projects at universities and companies, and worldwide democratized red teaming

- **but what makes for a great demonstrator for secure hardware?**

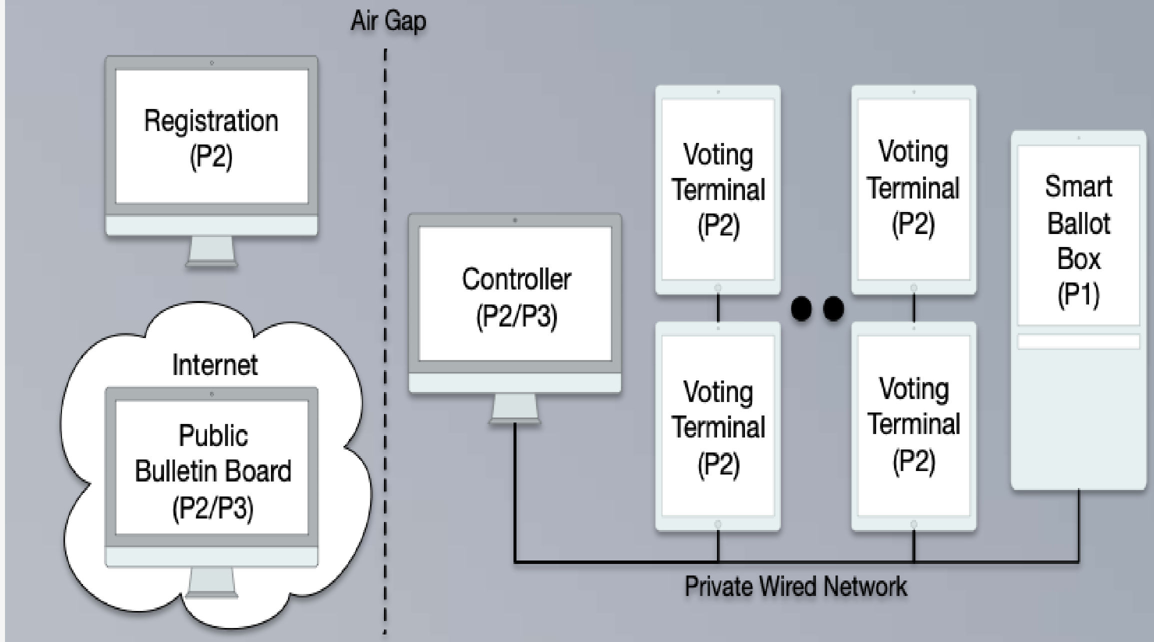
PROPERTIES OF A GREAT SECURE HARDWARE DEMONSTRATOR

- ☑ moderately complex domain
- ☑ not a classified system
- ☑ understandable to the public
- ☑ topical wrt today’s adversaries
- ☑ representative of nationally critical infrastructure and DoD systems
- ☑ needs all three sizes of CPUs
- ☑ interesting to the media and public
- ☑ security properties must intersect all seven vulnerability classes
- ☑ public good impact as a side-effect
- ☑ **ideas, technology, and execution must be relevant to DoD systems**

THE DEMONSTRATOR FOR SSITH SECURE HARDWARE
AN OPEN SOURCE, OPEN HARDWARE, HIGH-ASSURANCE VOTING SYSTEM

- election technology…
  - is on everyone’s minds
  - is nationally critical infrastructure
  - is notorious for security flaws

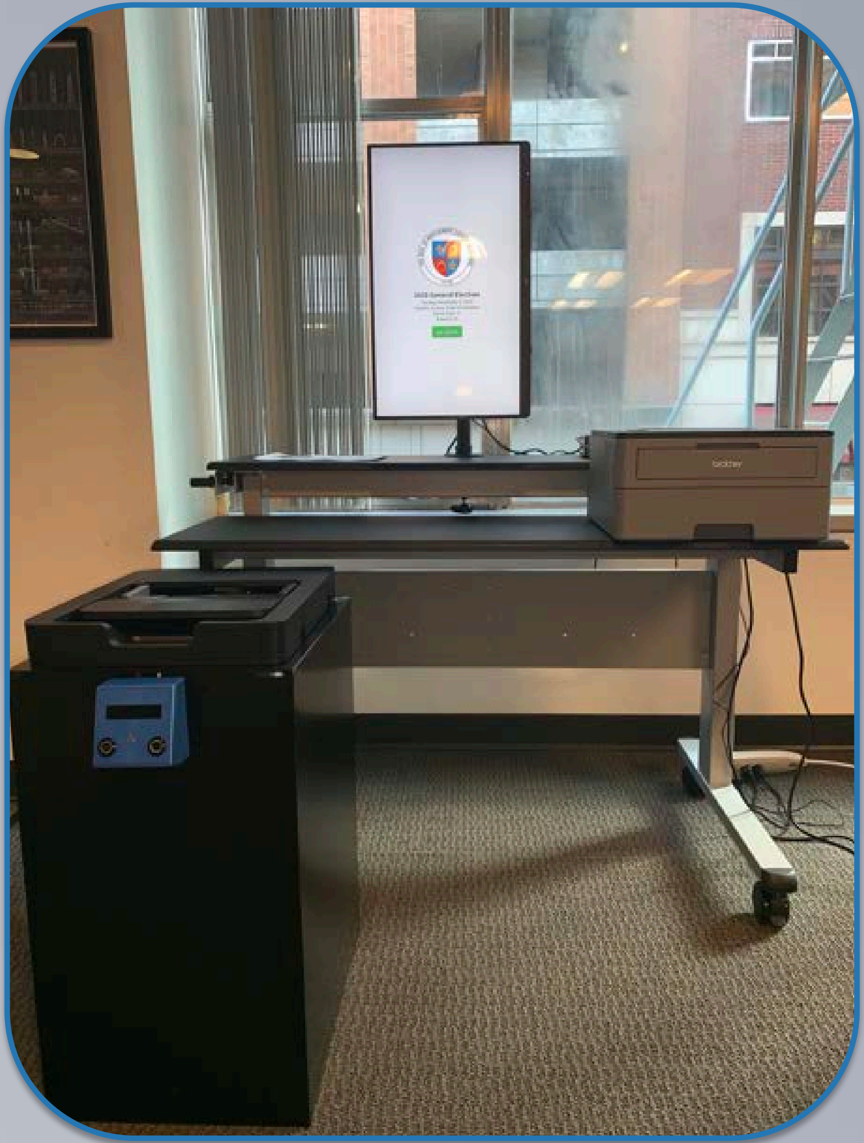
- a modern voting system…
  - needs a microcontroller (in the ballot box that accepts paper ballots),
  - a desktop CPU (for pollbooks, ballot marking devices, and hand-marked paper ballot scanning), and
  - a superscalar CPU (for tabulation and reporting evidence to the public)
  - must be open hardware and software



DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

THE DEMONSTRATOR FOR SSITH SECURE HARDWARE
AN OPEN SOURCE, OPEN HARDWARE, HIGH-ASSURANCE VOTING SYSTEM

- election technology…
  - is on everyone’s minds
  - is nationally critical infrastructure
  - is notorious for security flaws
- a modern voting system…
  - needs a microcontroller (in the ballot box that accepts paper ballots),
  - a desktop CPU (for pollbooks, ballot marking devices, and hand-marked paper ballot scanning), and
  - a superscalar CPU (for tabulation and reporting evidence to the public)
  - must be open hardware and software



DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

THE DEMONSTRATOR FOR SSITH SECURE HARDWARE
AN OPEN SOURCE, OPEN HARDWARE, HIGH-ASSURANCE VOTING SYSTEM

- election technology…
  - is on everyone's minds
  - is nationally critical infrastructure
  - is notorious for security flaws

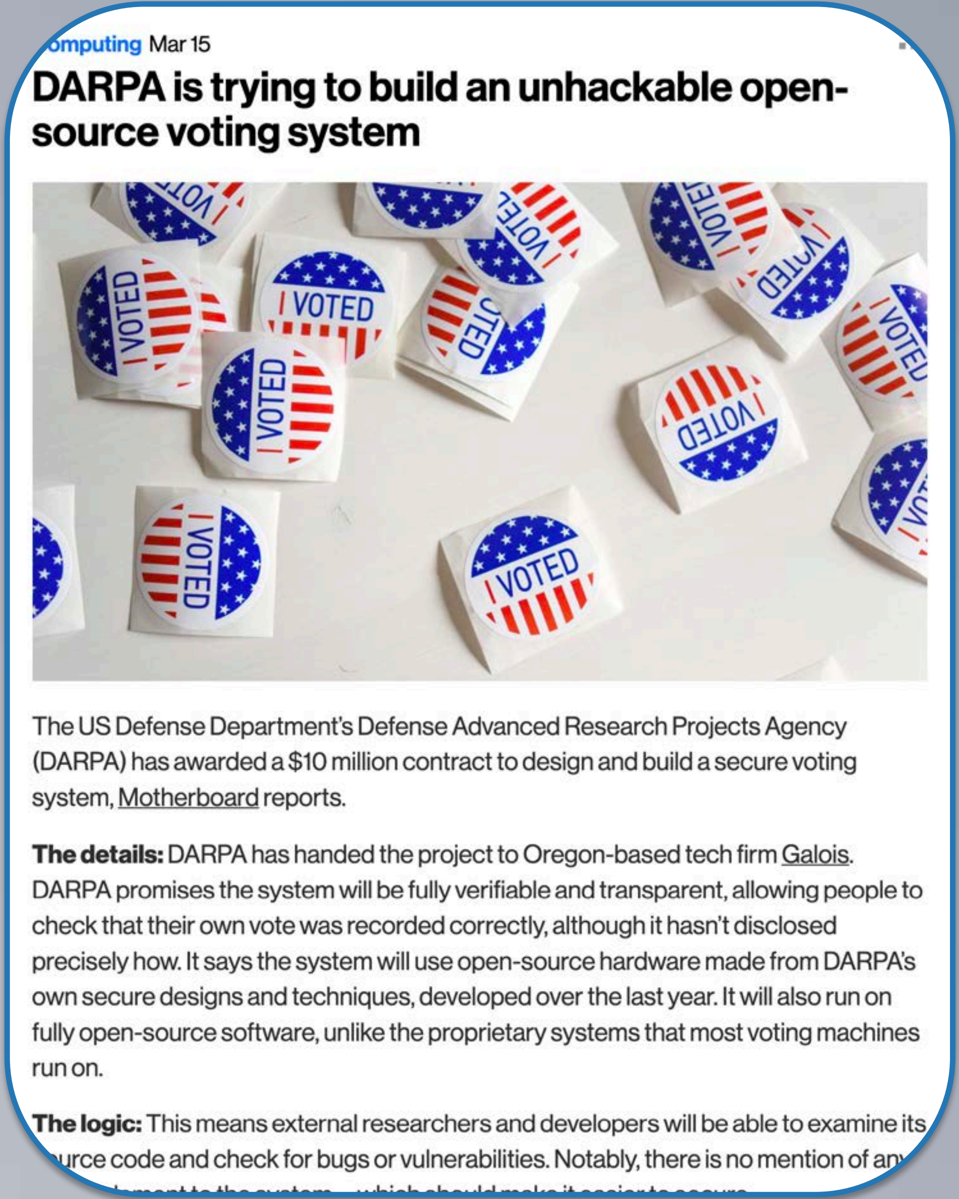
- a modern voting system…
  - needs a microcontroller (in the ballot box that accepts paper ballots),
  - a desktop CPU (for pollbooks, ballot marking devices, and hand-marked paper ballot scanning), and
  - a superscalar CPU (for tabulation and reporting evidence to the public)
  - must be open hardware and software

DISTRIBUTION STATEMENT A: Approved for Public Release,
Distribution Unlimited

COMMUNITY REACTION

- over the past 20 years we have developed relationships with many parties relevant to voting…
    - federal, state, and local agencies & officials (EAC, NIST, DHS, NASED, NASS, iGO, etc.)
    - election technology vendors (ES&S, Hart-Intercivic, etc.)
    - elections integrity organizations (Verified Voting, Brennan Center for Justice, EVN, etc.)
    - the media (print & filmmakers)

The enthusiasm for this demonstrator has been remarkable!



omputing Mar 15

DARPA is trying to build an unhackable open-source voting system

The US Defense Department's Defense Advanced Research Projects Agency (DARPA) has awarded a $10 million contract to design and build a secure voting system, Motherboard reports.

The details: DARPA has handed the project to Oregon-based tech firm Galois. DARPA promises the system will be fully verifiable and transparent, allowing people to check that their own vote was recorded correctly, although it hasn't disclosed precisely how. It says the system will use open-source hardware made from DARPA's own secure designs and techniques, developed over the last year. It will also run on fully open-source software, unlike the proprietary systems that most voting machines run on.

The logic: This means external researchers and developers will be able to examine its urce code and check for bugs or vulnerabilities. Notably, there is no mention of an

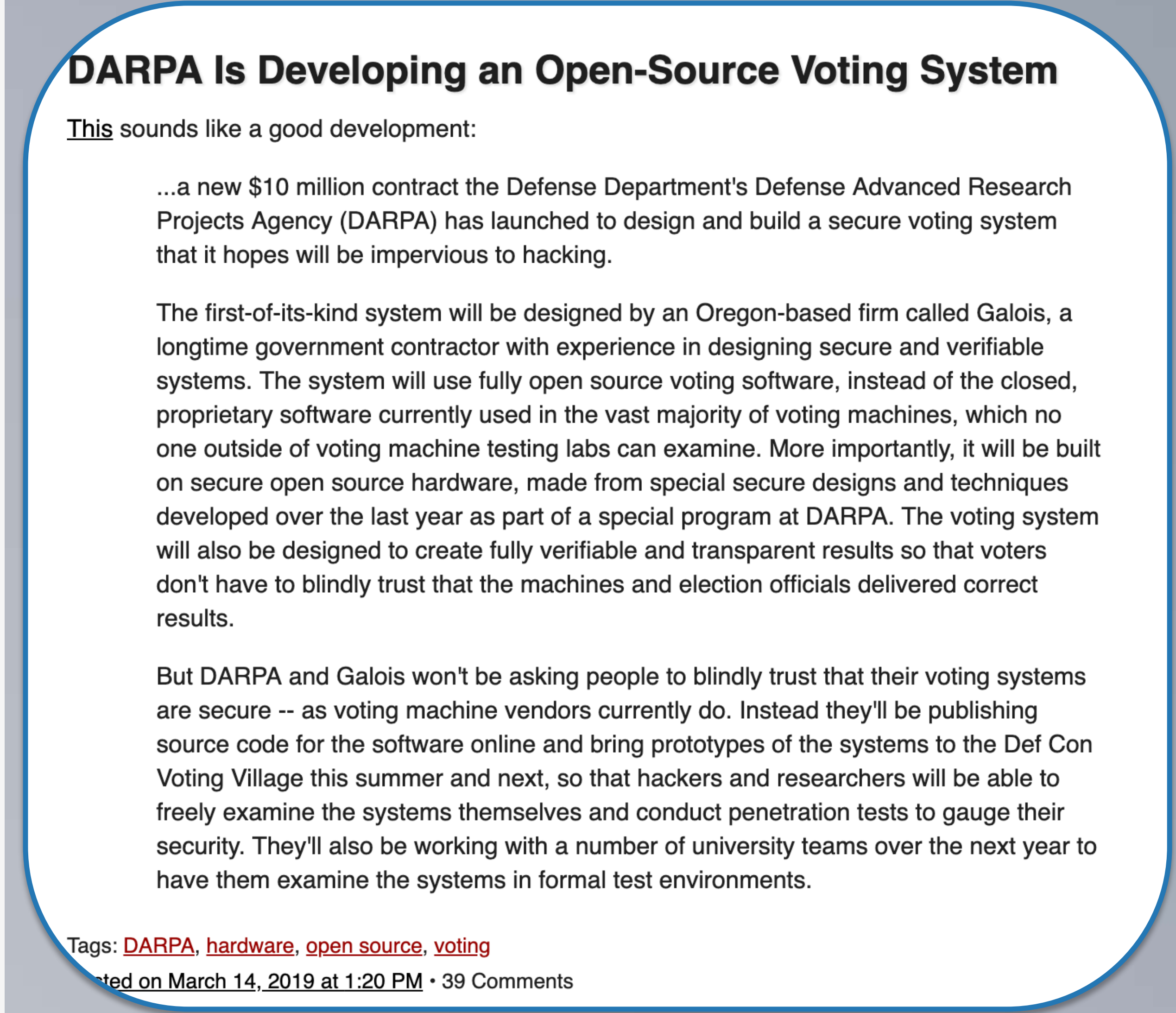
MIT Technology Review

DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution Unlimited

COMMUNITY REACTION

- over the past 20 years we have developed relationships with many parties relevant to voting…
  - federal, state, and local agencies & officials (EAC, NIST, DHS, NASED, NASS, iGO, etc.)
  - election technology vendors (ES&S, Hart-Intercivic, etc.)
  - elections integrity organizations (Verified Voting, Brennan Center for Justice, EVN, etc.)
  - the media (print & filmmakers)

The enthusiasm for this demonstrator has been remarkable!



DARPA Is Developing an Open-Source Voting System

This sounds like a good development:

> ...a new $10 million contract the Defense Department's Defense Advanced Research Projects Agency (DARPA) has launched to design and build a secure voting system that it hopes will be impervious to hacking.
>
> The first-of-its-kind system will be designed by an Oregon-based firm called Galois, a longtime government contractor with experience in designing secure and verifiable systems. The system will use fully open source voting software, instead of the closed, proprietary software currently used in the vast majority of voting machines, which no one outside of voting machine testing labs can examine. More importantly, it will be built on secure open source hardware, made from special secure designs and techniques developed over the last year as part of a special program at DARPA. The voting system will also be designed to create fully verifiable and transparent results so that voters don't have to blindly trust that the machines and election officials delivered correct results.
>
> But DARPA and Galois won't be asking people to blindly trust that their voting systems are secure -- as voting machine vendors currently do. Instead they'll be publishing source code for the software online and bring prototypes of the systems to the Def Con Voting Village this summer and next, so that hackers and researchers will be able to freely examine the systems themselves and conduct penetration tests to gauge their security. They'll also be working with a number of university teams over the next year to have them examine the systems in formal test environments.

Tags: DARPA, hardware, open source, voting

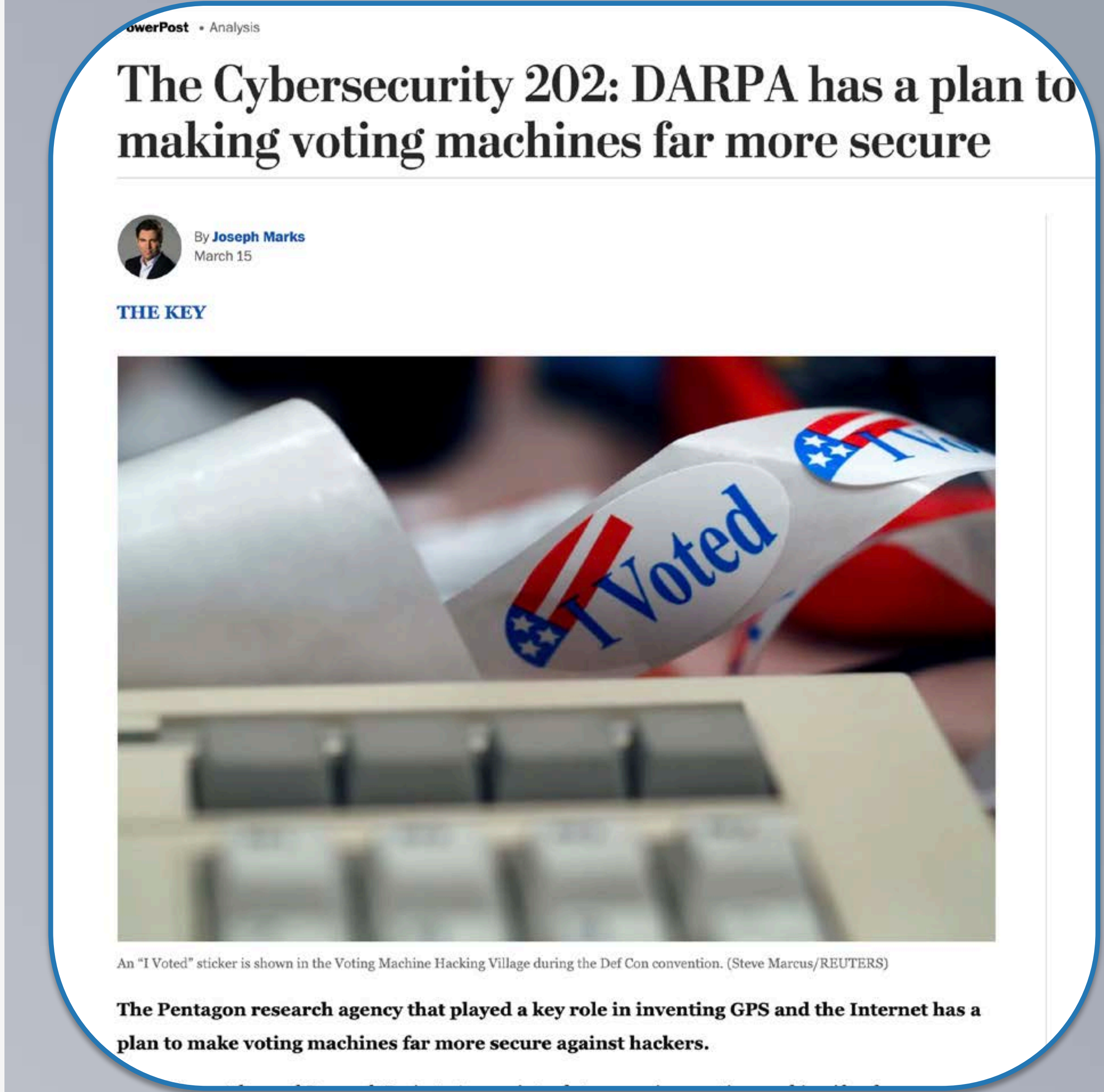
sted on March 14, 2019 at 1:20 PM • 39 Comments

Schneier on Security

COMMUNITY REACTION

- over the past 20 years we have developed relationships with many parties relevant to voting…
  - federal, state, and local agencies & officials (EAC, NIST, DHS, NASED, NASS, iGO, etc.)
  - election technology vendors (ES&S, Hart-Intercivic, etc.)
  - elections integrity organizations (Verified Voting, Brennan Center for Justice, EVN, etc.)
  - the media (print & filmmakers)

The enthusiasm for this demonstrator has been remarkable!



owerPost • Analysis

The Cybersecurity 202: DARPA has a plan to making voting machines far more secure

By Joseph Marks
March 15

THE KEY

An "I Voted" sticker is shown in the Voting Machine Hacking Village during the Def Con convention. (Steve Marcus/REUTERS)

The Pentagon research agency that played a key role in inventing GPS and the Internet has a plan to make voting machines far more secure against hackers.

Washington Post

SSITH'S IMPACT: SSITH WILL…

- show the world that **critical infrastructure** and **national security systems** can be **transparent, correct, and secure**
- create a **case study in layered security using formal methods**
- show that, to tackle these challenges, **formally assured software, firmware, *and* hardware is mandatory**
- **influence perceptions, planning, and strategy of corporations**
- **bring hardware engineers security super-powers**
- boost their security capabilities **without making them experts in cybersecurity**

- ***SSITH will help entities that are creating silicon and electronic design automation tools tackle 21st century security so that we can all benefit.***

The MORPHEUS chip from the UM and UT Austin represents a very interesting approach, and you'll hear about it next from Prof. Todd Austin.

FOR MORE INFORMATION

- On Twitter…
  - https://twitter.com/galois @galois
  - https://twitter.com/free_and_fair @free_and_fair
  - https://twitter.com/votingvillagedc @votingvillagedc
- On GitHub…
  - The BESSPIN Voting System
    https://github.com/orgs/FreeAndFair
  - The SSITH RISC-V cores, OSs, compilers, etc.
    https://github.com/orgs/GaloisInc
- On the web…
  - https://galois.com/ and https://freeandfair.us/

ERI ELECTRONICS RESURGENCE INITIATIVE

SUMMIT

2019 | Detroit, MI | **July 15 - 17**