National Cyber Leap Year Summit 2009:
Exploring Paths to New Cyber Security Paradigms
Draft Report of Participants' Ideas

August 24, 2009

New Game: Knowing when we've been had.

This document explores Hardware-Enabled Trust as a path to this new game.

The following ideas were captured in unedited form at the National Cyber Leap Year Summit. The ideas are a summary of the discussion of the participants in the Hardware-Enabled Trust session. They do not necessarily represent the opinions of the co-editors or the organizations they represent. The Summit is managed by QinetiQ North America at the request of the NITRD Program, Office of the Assistant Secretary of Defense Networks and Information Integration, and the White House Office of Science and Technology Policy.

Please **provide your comments**, if any, **by September 3, 2009** for utilization by the Summit's program co-chairs at http://www.co-ment.net/text/1449/. To add a comment, select the "Add" tab in the left navigation menu, select (highlight) the portion of the document you are commenting on, and provide your comment. If commenting on an entire section, you may select the section heading to anchor your comment.

If you have any further questions or comments, please visit the National Cyber Leap Year Web site at the following address: http://www.nitrd.gov/NCLYSummit.aspx, or send email to leapyear@nitrd.gov.

What is the new game?

One of the hardest things about today's game is not being aware when we're losing. Our trusty PC has no way to notify us that it has in fact become an enemy agent or a zombie, secretly exfiltrating our financial secrets to identity thieves, or spamming our neighbors for some botmaster. Since we have no real plan for checking and restoring the integrity of our assets once we start using them, we are forced into the impossible position of having to deploy impregnable systems. In the new game we persistently monitor our assets for changes in trustworthiness by embedding tamper-resistant roots of trust in the architecture. Attacks can be stopped in their tracks if we can isolate and decontaminate their host.

1 Introduction

There was no attempt to provide comprehensive coverage of all the ideas in the areas of hardware-enabled trust. The list above is a simplified categorization of the product of a brainstorming session. Below is a snapshot of the discussions of Group 5, covering most of the topics discussed during the session. Some of the ideas discussed in this appendix are covered in more detail in the Chairs' report.

Seven (7), ten year long-term goals were initially identified from which ideas were identified and put into seven (7) categories. These ideas were subsequently regrouped into six (6) and finally four (4) ideas. The distillation process is discussed following the description of the four (4) final game changing ideas.

The group developed action plans for the focus areas and in the process revised the focus areas to include:

- End to End Trust
- Hardware defenses for attacks
  - Hardware that does not leak
  - Hardware monitoring of normal behavior
- Resilience
- Secure Cloud Storage

A general purpose action plan strategy is:

- Institute a competition for building the best secure widget
- And a competition to break it

Common aspects of action plans:

- Develop national security standards for testing hardware
- Competition (as described above)
- Industry-academic teams are key to success

2 End to End (e2e) Trust

2.1 Description

- Need minimum (canonical) set of trust properties, protocols to exchange them, trust infrastructure to support these operations
- The canonical set may include:
  - e.g., Secure key management
  - e.g., verifiable identity, attestable identity
  - Ability to contain or isolate

- What are these trust properties? (a research question)? They need to be defined
- Domain-specific abstractions are necessary for the big picture
- (Research question) Object-oriented extensible language with defined operations, supported by hardware

- Other important considerations:
  - Interoperability is key to end-to-end trust
  - Hardware based protection of audit ability
  - Heterogeneous systems, from sensors to servers, need to be able to enforce trust in a uniform fashion
- Secure the e2e trust in distributed heterogeneous environment
  - Including storage, computation and communications
  - Devices need canonical set of properties supplemented with domain specific identity
  - Privacy preserving identities
  - Identify who/what we deal with.
    - Where we want to go (what devices/services/networks we want to access)
    - Set of principles describing the environment
- Observations
  - Some level of anonymity is possible in some areas but not everywhere (e.g. it is limited in cell phone networks)
  - NetBooks may be a tipping point – trust techniques may start there
  - Signatures of software components are more available than the same information on hardware components. SignaCert has over 600,000,000 signatures of software

2.2 Inertia

- Infrastructure takes time to build
- A forum is not an efficient enabling mechanism
- Need legal support because of the nature of the problems - Need private sector participation
- Public does not understand the threats; outreach is necessary
- Cannot define definitive minimal set (no agreement on this)
- Application writers, system developers don't even use TPM that is widely available; will they use the new generation of trust technologies?
- Cannot test what is not specified by manufacturer, e.g., hardware backdoors/Trojans inserted by hardware designer

2.3 Progress

Other possibilities

- Reduce barrier to acceptance by creating (inter)national authoritative repository for whitelist component signatures (www.isitsafe.org) - Software and hardware
- Create a compromised or revoked key service

- Define bottom-up abstractions – e.g., domain-specific objects with allowable operations
- Possible to define generic canonical sets of security properties, e.g., secure storage of long-term secrets (e.g., private keys), but rest is domain-specific
- Data-sheet (retrievable) of functionality provided by hardware components (Component provenance data-sheet) may be part of trust information, already exists. Data sheets need to be portable across heterogeneous test systems
- Set up environment where people will use provided hardware security features
- Need new business model to make a difference implementing suggested changes

2.4 Action Plan

- Determine canonical set of common health properties supplemented by domain provided information that systems should be able to request and attest
  - Supporting on demand health checks
  - Supporting dynamic measurements
- Develop a trusted unforgettable identity down to the component for devices/platforms
  - To allow for correct attribution
  - To manage connections
  - To manage and enforce trust
- Study and determine the market drivers that create demand for trust? Study why TCG concepts have yet to make traction;
  - Where is the Velcro holding things back
  - What incentives are possible to change the situation?
- Create a national trusted infrastructure test bed - Government, academia and industry participate
- Identify and develop standards for component and device identification
  - Create the DNA to describe a system top to bottom starting at the IC level
  - The information can be used for attestation (as pass/fail), but not disclosed for privacy reasons

2.5 Jump-Start Plan

Establish an operational pilot implementing these concepts including infrastructure to enable remote attestation (short term TCG-based; subsequently next generation of trust technologies)

- Users Group to take specifications to implementations for market segments - Verticals application domain possibilities need to be explored and facilitated Financial, Health Care, SCADA
- Sponsor a forum for verticals to collaborate and identify common infrastructure needs – Standards for inter-trustability need to be developed

3 Enable Hardware to Counter Attacks

- Information leakage thru side-channel, covert channel attacks
- Attacks connected with physical possession of device
- New hardware features for performance that degrade security, e.g., cache, Hyperthreading

3.1 Description

- Considers both hardware and software attacks
- Hardware defenses for:
  - Information leakage due to hardware -induced Side channel channels
  - Continuous measuring normal behavior
    - Robust characterization of normal behavior
    - Hardware Trojans – can develop state machine for hardware system for normal behavior
    - Hardware to protect measurements
    - Hardware to do monitoring (like IBM service processor)
    - Sanitization features for malware already present
    - What we measure
- Hardware verifies system integrity at runtime
  - Continuous biometrics, continuous monitoring
  - Dynamic measurement

3.2 Inertia

- Costly, chip yield is limited, severe performance degradation
- Software attacks are prevalent
- Hardware attacks had been kept as classified by governments; no good source of information is available.
- Definitions are needed:
  - Health: (need definition)
  - Integrity: hash-identity, safety (device does not blow up),
- Overhead of storage of this additional information – from monitoring and metrics-- is proportional to cache line (7-14% more storage), for those systems that already do this

3.3 Progress

- hardware cost decreases, while computers are being used for more important transactions, data and control of critical infrastructures
- Mobility increases the risk of hardware attacks
- Cloud computing (servers) magnifies the effect of the attacks; hardware capability would be helpful.
- We look to reliability for new ideas (e.g., N-version programming)?
- Aspects of system's integrity/health that can be characterized as system evolves

- We will be able to measure integrity of high-assurance software. We will be able to determine the cause of corruption/security issues (how did software get to that corrupted state?)

3.4 Action Plan

Elevate the importance of security in the design of hardware performance and power features. Make security a 1st class citizen in hardware design

3.5 Jump-Start Plan

4 Enable hardware to counter attacks—Hardware that does not leak hardware defenses for information-leakage attacks (side-channel attacks)

4.1.1 Description

Example: software cache-base side-channel attacks

- Memory leak problem (garbage collection)
- Software cache-based side-channel attacks

4.1.2 Inertia

- Security has not been considered important enough.
- Hard to enumerate the security properties
- Non-leaking versus alternative implementations needs to be considered
- Intellectual property issues will arise

4.1.3 Progress

- Possible if government can do something

Other discussions

- Identify a method to fingerprint hardware so it can be vetted
- Interoperability of test data

4.1.4 Action Plan

Short term

- Try hardware solution for secure and high performance cache
- Figure out metrics for side channel attacks: how do we measure the severity of side channel attacks?
- How do we quantify the risks: We need to understand how to quantify the risks associated with attacks

- Figure out metrics for evaluating the security properties of a design: What should designers be looking for? How do they evaluate design options?
- Prototype designs that have already been proposed.

Long Term

- Design secure hardware subsystems that are both secure and high performance
- Establish a set of criteria that represent acceptable levels of security: The objective is to give guidelines to improve designs over time.
- Set of design principles for secure processors:
- Establish a method to verify hardware integrity:
- Roadmap for improving identifiable metrics:

4.1.5 Jump-Start Plan

- Read proposals and prototype, and give feedback if it works
- Establish competition (open collaborative teams come up with design) design competition then break competition. - Use open-cycles government has in trusted fabrication labs

5 Enable hardware to counter attacks—Continuous hardware monitoring of normal behavior

5.1 Description

- Hardware can automatically collect data and may be non-by-passable
- Hardware can protect measurement data and procedure

5.2 Inertia

- Serious data bandwidth is necessary to collect data
- Has been done already – in networks and in software
- Has problems in identifying legitimate behavior
  - False positives
  - False negatives – may miss problems

5.3 Progress

- Multicores allows parallel monitoring
- Incentivize manufacturer to join hardware fingerprinting efforts
- Run competition
- New in computing devices

5.4 Action Plan

- Short term

- identify measurement technology that can measure normal behavior of software and hardware systems
- Measure low-entropy systems, e.g., web-server and a SCADA system - Can add to SCADA system
- See if it handles legitimate peak loads - Chron tab (irregularly scheduled jobs or activities)
- Methods for Process calibration
- Identify what can be measured
- Making mounds of data about program behavior available

- Long term
  - Identify what should be measured to characterize multicore behavior, user behavior
  - Methodology for applying what the parameters are for measurement
  - Hardware collection of behavior

5.5 Jump-Start Plan

6 Resilience

6.1 Description

Commodity hardware still executes critical services even when compromised

- Tools
  - Redundancy
  - Diversity
  - Checkpointing / roll-back
  - Reconfigurability / self-repair / evolution
- Instantiation
  - Multi-core processors

6.2 Inertia

- Hierarchical trust model - Full-stack attestation (TPM) – the operating system architecture has severe limitations
- To get full effect from hardware diversity, need more software diversity – extra complexity does not improve security - Diversity may add more attack vectors (want vertical rather than horizontal diversity)
- We will never get vulnerability-free software, but execution of malware is the problem

Challenges

- Industry buy-in
- Costs (area, power, complexity, design, validation, etc.)

- Lack of incentives
- Integration of different techniques
- Hard to create meaningful scoring systems for security (resilience or up-time easier), - What are the set of general properties that must be tested?
- Hard to get vendors to move up on the Evaluation scale
- Done previously
- Must be unobtrusive, not hog battery, performance

6.3 Progress

- Processors are cheaper with multicore/manycore - Operating systems are getting better about incorporating hardware features
- We compute on many computers/devices, and attacker has to break many to get to protected data
- We can use diversity to improve security
- Open source may improve diversity techniques
- Can leverage reliability mechanisms (at some stronger level with some changes) to provide greater resilience to attacks
- NISB could apply to testing other trust properties - May need parallel board like NTSC and FAA – testing versus enforcement organization, but may be single organization
- Metrics – passes which set of sets (stars)
- Sets of tests like the EU's randomization sets (Estream)

6.4 Action Plan

- Establish benchmarks / define scope
- Program to build a prototype with commodity components (1yr/$1M) - Platform for experimentation. Develop tool kits.
- Funding for research and prototypes (5yr/$50M) - Academic + industry teams to build a system prototype
- Establish a National Information Safety Board (NISB, a federal evaluation / test organization) Note: this entity was renamed in chairs report.
- Security standards for federal purchases

Discussion

- Academic + industry teams are expected to integrate different tools to build a system that can meet the standards set by benchmarks for the government program
- NISB will test all commodity systems and publish the test scores. - Consumers will be encouraged to buy systems with high scores just as they are encouraged to buy cars with higher crash test scores from NTSB.

6.5 Jump-Start Plan

7 Trustworthy Storage and Data

7.1 Description

- Self-protecting data is the way to proceed.
- Deployable key management solutions must be built into hardware of commodity products.
- Develop Prototype Secure Storage Area Network (SAN) System
  - SAN controllers are no longer disk drives hiding behind server systems
    - Proposal is not limited to a strict definition of SAN
    - Network attached storage is also included here
  - Full fledged network nodes (using iSCSI)
  - Vulnerable to full class of network attacks
  - Need to selectively share data with multiple clients with different security properties
- User controls needed
  - Access controls need to be both mandatory and discretionary
  - Data owners need to be able to specify policies
  - Mandatory controls needed to control malware

Types of Data

- User-control of data is important
- Grey data
- Self-describing data

7.2 Inertia

Why hasn't this been done?

- Storage designers are mostly disk designers – they view security as a problem for the server – NOT for them
- Initial version of object store didn't address limiting capability propagation
- Why bother – implement and prove object-oriented self-describing and protecting data – then let people/vendors catch up
- Unanticipated use of various technology and their confluence – perfect storm
- Cloud storage – cool – may lead to horror reality – need horror story examples
- No incentive for industry to collect problem - Need economic motivation, but even this may not be enough
- Main motivators: FEAR and AVARICE
- Public awareness of security risks

What would derail the change?

- Controllers not sufficiently resistant to network attacks
- SAN's security is not as good as previous disk farms, and needs built-in security

- Key-management is difficult for data at rest
- Cryptography hard to implement correctly, especially in distributed environments
- Data leakage protection –covert channels hard to stop

7.3 Progress

Technically Feasible

- Mechanisms for secure SCSI disks may also solve Side-channel attacks
- SCSI standards committee – object Store w/ capabilities
- (TPM-std) encrypted disk drives for short term, IEEE std 1667 are improvements
- Object-oriented architecture
- Context where objects can be viewed
- Crypto well developed, key-management solutions are studied; a lot of work done in this area.

Environmentally Feasible

Existing standards work ongoing in this space

- TCG has standard for disk storage
- Object Store is a capability-based standard under development by storage community
- Existing cryptographic standards for key protection (didn't catch the standard number)

7.4 Action Plan

Joint Academic/Industry project to build and demonstrate a SAN controller to defend against all of these classes of attacks

Timeframe

- RFI in 60-90 days
- RFP 9 months after that
- 2-3 years contracts
- Need both academic and industrial team members
- Need to ensure competitiveness – standards need to be open – don't let one company lock itself in
- Implementations could be open or proprietary

Multiple approaches

- The RFP should permit multiple approaches and multiple contracts should be considered
- Some could be based on securing the SAN controllers
- Others could be based on encrypting the data before the SAN controllers ever see it

7.5 Jump-Start Plan

See 60-90day implementation above

7.6 Comments

Game change by the bad guys

- Targeted spear fishing attacks to steal data
- Data Leakage Protection (DLP) products under development by various companies
- DLP products are re-discovering confinement and information flow control
  - Some developers are unaware of the extensive work in mandatory access controls dating back to the 1970s
  - Others are well aware
- As DLP products get established, commercial covert channel attacks will become common
  - Bad guys are well aware of covert channels
  - Haven't used them much YET, because other attacks were easier
  - But as DLP products become effective and widely available, the bad guys will be quite capable of using covert channels

8 History of Idea Development

As noted previously, the group first identified long-term goals and grouped them into seven (7) categories and ultimately focused on four (4) broad, encompassing ideas as outlined in sections 1 through 5.

8.1 Leap-ahead, Long Term Goals – 10 year

1. We will build a computer that will not execute malware
2. We will be able to make a determination whether to trust a device, a network, or a software package based on dynamically acquired and exchanged standard trust information and user defined trust and security policies
3. A user will be able to make an informed decision about purchasing a device or a service based, in part, on independent security scoring.
4. Transactions will be dynamically re-routed into an optimal trusted path, independently from their origination in terms of device, network, and application.
5. Distributed data objects will be able to protect themselves based on minimum security sets and user defined policies.
6. Security will be considered a core feature when architecting hardware.
7. New trust models will be introduced that are rooted in hardware instead of enforcing hierarchical interdependencies across the software stack

8.2 Initial Ideas

Ideas were generated and grouped into categories.

1. New trust models enabled by hardware (substituting hierarchical models with hardware-rooted models with fewer inter-dependencies)

2. Resilience as a foundation for security features
3. Hardware defenses for hardware attacks
4. Evaluation and dynamic measurement
5. End-to-end trust in a heterogeneous environment, in order to enable end-to-end communications assuring an acceptable level of security in a heterogeneous (diverse networks and devices, from sensors to servers) and distributed (e.g. cloud computing) environment
6. Trustworthy storage and data rooted in hardware
7. Designing crypto/randomization into core computer hardware in order to support secure execution and secure storage

8.2.1 Idea Development

1. New trust models enabled by hardware – what does it mean? List of ideas.

- How can hardware protect trusted applications and data? - Even if the operating system is compromised
- Replace hierarchical trust models with alternative solutions
- Hardware provides essential security in systems
- Object-oriented representation of data and associated allowed operations and constraints can serve as basic architecture, permitting us to build the system bottom up
- Start from scratch with a clean slate to see what can be done - The results can potentially be retrofitted into existing systems, but often it is hard to incorporate the best ideas into existing architectures)
- Hardware decoys with low overhead can be incorporated into the standard set of security activities
- Hardware-enabled trust must be very low cost, and take into account short lifetimes of commodity hardware
  - How about very thin simple hardware clients?
  - Take all intelligence out of hardware to make it simpler

2. Resilience as a foundation for security – what can be done? List of ideas.

- Take advantage of diversity and redundancy, e.g., for resilience and security
- Hardware built with resiliency and fast recovery mechanisms will improve overall security
- We need hardware that can run security-critical tasks even if the system is partially corrupted
- Take advantage of distributed computing platform – spread spectrum computing

3. Hardware defense for hardware attacks – what can we do? List of ideas.

- Enable hardware to counter hardware attacks
- Design hardware that spots and counters malicious hardware - Must allow legitimate upgrade and replacing of hardware components
- Recognize, measure and enforce normal (not abnormal) behavior—the set of normal behavior is far smaller than the set of abnormal behavior

- Continuous measurement, can track by user ID or other, more privacy-conscious, parameter
- Ensure there is a way to know what is "your" hardware

- Commodity-level tamper resistance

4. Evaluation –having implemented security features, we need a reliable way to evaluate them. List of ideas.

- Practical security evaluation of hardware, e.g., ability to attribute to manufacturer, ensure authenticity
- Tools that verify that the product of fabrication is correct with respect to the specifications sent to fabrication
- Evaluation process that is open and reproducible
- Both the design and each instance
- Third party and self-evaluation

5. End-to-end trust in heterogeneous and distributed environments

- Need minimum set of trust properties, protocols to exchange them, trust infrastructure to support them (all networks, all devices, from sensors to servers)
  - Interoperability
  - Hardware-based protection of auditability
- Use hardware to assess identity and health of systems
  - E.g., Continuous biometrics
  - Continuous measurements
  - Hardware verifies system integrity at runtime
- Secure hardware interfaces
- Operating systems leverage TPM and future trust features rooted in hardware
- Considerations raised by cross-group synthesis discussions with other groups:
  - Community situation awareness and a shared ontology are necessary to convey and implement a shared version of trust
  - Distributed defense based on behavior-based models will be helpful
  - Transient dynamic communities of trust will emerge; need to be considered as systems are built
  - How can hardware support whitelisting of software that runs on it?
  - Hardware can provide acceleration of world-switching (VMs)
  - Hardware-enhanced accountability is a useful notion (e.g., ensure attribution and non-repudiation)
  - How can hardware help detect an insider attack? Possibilities exist.
  - Can hardware provide a set of data about suspicious activity?
  - Automatic reporting for national repository of suspicious activity
  - Incentives for active defense (e.g., in hardware)
  - A repository of patterns of communication (assisted by hardware collection)

- Multi-layered defense
- Hardware-assisted continuous ground-truth evaluation
- Different levels of service provision

6. **Trustworthy storage and data – what can be done? List of ideas.**

- Secure cloud storage needs to be controlled by user
- Do not restrict the flow of data, restrict the interpretation of data instead
- Use hardware to provide auto-redaction (minimization, anonymity, sanitization) to handle flow of information to protect lives, privacy, etc.; Ensure removal of sensitive data
- Hardware performs provenance checking
- E.g., to recognize good code
- Even if the operating system is compromised (feedback from data provenance group)
- Need attributes defined in order to perform attribution and checking
- Protection of provenance information itself can be rooted in hardware
- A research question: How to architect a coherent secure data and storage system, e.g., Cloud, using developed ideas to achieve practical and resilient design. Combine industry and academia in a single team.

7. **Designing crypto/randomization into core computer hardware for secure storage and secure execution – List of Ideas**

- Can crypto improve availability? Crypto is a great tool for improving confidentiality and integrity. But are there new crypto techniques that can help improve availability, resilience?
- Is there a field of math, e.g., randomization theory, that can help improve both security and performance? Such an approach will help: thwart attacks while improving performance
- We need to bring mathematicians (crypto, randomization) and computer architects together? (process)
  - How to get effectively “no-overhead” crypto?
  - How to use crypto and randomization to advantage in new environments, e.g., in processor pipelines, multicore

8.3 Focus Areas

A set of ideas was selected by the co-chairs for more detailed examination.

- End-to-end trust
- Use hardware to assess identity and health
- Enable hardware to counter hardware attacks
- Resilience as a foundation for security
- Trustworthy storage and data
- Crypto and randomization in processors and memory

8.4 Game changing Ideas

The group developed detailed plans for the four (4) game changing ideas discussed in sections 1 though 5 and in the process revised the focus areas to include:

- End to End Trust
- Hardware defenses for attacks
  - Hardware that does not leak
  - Hardware monitoring of normal behavior
- Resilience
- Secure Cloud Storage