NICIAR: Disruptive Technologies for Information Assurance

Carl E. Landwehr
Program Leader, NICIAR

Carl Landwehr
NICIAR Program Leader
Intelligence Advanced Research Projects Activity (IARPA)
301-226-9100
email: CLandwehr@iarpa.org
New DNI, Mike McConnell:
• Intelligence Community Integration
• Acquisition emphasis
• Information sharing
  • Need to know vs. responsibility to provide
• Analyst at the center
  • Know the customer needs
  • Know the sensors and source
What is “disruptive technology?”

Technology that precipitates change

Examples:
- Digital vs. chemical photography
- Packet vs. circuit communications
- Wireless vs. wired telephony
Research Areas

- National Intelligence Community Information Assurance Research
- Quantum Information Science and Technology
- Exploratory Investigations
- Information Exploitation
- Research Development and Experimental Collaboration (RDEC)
Vision:

Level the cybersecurity playing field
- Dramatically improve the fundamental trustworthiness of the NIC cyber infrastructure
- Defend existing NIC cyber infrastructure from external and internal threats; enable operation despite attacks

Goals:
- Use accountability as a lever to reduce vulnerabilities and foster information sharing
- Increase the attacker’s cost to penetrate NIC systems
- Provide usable and flexible security mechanisms

Defense has an uphill battle!

Flawed software
Spoiler network protocols
Complex security management
Niciar Research Topics

Accountable Information Flow

Goals:
• Incorporate accountable information flow mechanisms at all system layers
• Develop and demonstrate network designs in which today’s attacks are engineered out

Technologies:
• Physical unclonable functions, secure coprocessors, static/dynamic analysis

Large Scale System Defense

Goals:
• Increase attacker’s cost
• Enable system operation during attack
• Improve system configuration assurance

Technologies:
• Dynamic, diverse programs and systems
• Configuration specification and verification
TARPA

NICECAP Planned Timeline

Topic areas:
- Accountable Information flow
- New focus area 10/07: Privacy Protecting Technologies
- Large scale system defense

BAA release 4/24/06
Round I Work begins 6/1/07
Updated BAA release 10/2/07
White papers due 11/2/07
Full Proposals invited 1/8/08
Proposals due 2/6/08
Contract negotiations begin 4/15/08
Awards made 7/15/08
Double attacker’s time/resource cost to compromise NIC systems through remote exploits
- Unmodified system as baseline
- Applications: reduce vulnerability windows in time (patch generation/installation, reconfiguration) and space (flaw/fault detection and removal)

Decrease by half the time and effort required to attribute a specific computational event/information flow to a (human/software/hardware) initiator
- Unmodified system as baseline
- Applications: sanitization, information sharing (credit), leakage (blame)

Stretch goal: Reduce by a factor of 10 the time/effort required to certify/accredit a new, conforming software component for use in a general purpose environment based on accountable information flow technologies
- Existing system and certification/accreditation process as baseline

Privacy goal TBD
**Large Scale System Defense**

*S. Bellovin, A. Keromytis, S. Stolfo, Columbia U.*

**PROBLEM:** Scalable Automated Collaborative Defenses

**Solution:** Artificially diversified detectors/software

- Leverage monocultures into distributed sensor net

Automated learning-based anomaly detectors

Automated patch generation: faster response

Better Patch Management: safe deployment

**PROGRESS**

Source code patching tool - DYBOC
- Stack and heap-based buffer overflow and underflow attacks

Emulator-based Inst. Set Randomization - STEM
- New content-based anomaly detector - Anagram
  - Experimental evaluation with R2

Data sanitization
- Model normal “attack-free” traffic

DNAD-2: (formerly called Worminator/Whirlpool)
- Content-centric Distribution System (sensor independent)
- Source IP’s of detected attacks *e.g.* Snort IDS alerts
- Aggregation of IDS alerts & reports
- Patches
ConfigAssure: Dynamic System Configuration Assurance for NIC Cyber Infrastructure,
S. Narain et. al., Telcordia / S. Malik, Princeton / D. Jackson, MIT

**APPRAICH**

- Specify security architecture via first-order logic constraints on component configurations.
- Use constraint solver to compute (a) component configurations implementing security architecture, and (b) plan to safely evolve to new configuration.
- Adapt infrastructure to preserve security architecture in response to contingencies.

**PLAN**

- Design ConfigAssure Architecture. (Nov 2007)
- Develop prototype and integrate Zchaff and Kodkod technologies developed by Princeton and MIT (May 2007)
- Conduct evaluations and trials (August 2008)
Goal:
Substantially reduce risk of misconfiguration of complex systems

Motivation:
Not only may a single non-compliant machine may pose nonlinear risk,
But unmanaged diversity of configurations with locally acceptable risks may aggregate into a set of total of risks that are not tenable.

Approach:
Enhance risk analysis by clustering large (to $10^5$ machines) configuration sets.
Employ Statistical analysis, visualization, and machine learning.
Monitor configuration from virtual partition
Employ tamper-protect embedded software agents.
Leverage/improve Relevant Open Standards (XCCDF, SCAP, OVAL)

Plan / Progress:
New start fall 2007
APPROACH:

- Restructure Software using Context-Free Grammar Transformations
  - Induce a context-free grammar from a program.
    - Construct the control-flow of a program from its executable.
    - Induce a context-free grammar from a control-flow graph.
  - Randomly transform the grammar.
    - Perform functionality-preserving context-free grammar transformations.
  - Construct a new program from the transformed grammar.
    - Deduce a control-flow graph from a context-free grammar.
    - Construct a program from a control-flow graph.

- Redistribute Parameters and Local Variables

PLAN

- Source code based demo (June 2008)
- Object code based demo (Sept 2008)
- Prototype tools (Dec 2008)
**APPROACH**

- Develop a unique single security mechanism that can defend against memory corruption exploits.
- Provide mechanisms for remediation, including diagnostics, recovery and repair.
- Make generalized defense practical by keeping run-time performance penalties small and making protections easily portable and scaleable.

**PLAN**

- Demonstrate ability to identify critical data and untrusted operations in binaries (Nov 2007)
- Demonstrate integrated policies using output of analysis phase (Jan 2008)
- Demonstrate additional policies (May 2008)
- Final prototype and supporting documentation (August 2008)
IFERRET: Improving Program Security through Traceable Dynamic Information Flow
T. Leek, G. Baker, R. Lippmann MIT-Lincoln Lab

Developers / security experts given ability to
• Monitor progress of sensitive and malicious information through and between programs
• Monitor many sources, allowing for more comprehensive security policies
• Ask, when a buffer overflow is about to happen, “Which parts of which inputs are responsible?”
• Ask, of a debugger, “What input bytes control this variable?”

IFERRET: Traceable Dynamic Information Flow System
– System-wide
– Traceback: Permits tracing any byte in memory back to parts of inputs from which it derives
– Unlimited sources
– Memory efficient: much less than one bit per byte
– Implicit Information Flow (IIF)

• PLAN (through 8/08)
  – Build IFERRET prototype, leveraging existing Taint Graph implementation
  – Evaluate
    – Track passwords from multiple sources
    – Track multiple local IP addresses
    – Trace back malicious input to one source among many
    – Measure miss and false alarm rates, overhead
**APPRAOCH**

- Develop a multi-variant code execution technique that will detect injected malware in real time and automatically repair affected code.
- Execute several slightly different instances of the same program in lockstep on multiple disjoint h/w processing elements
  - Uses system call randomization for code diversity
- A monitoring layer compares the computation states of the different instances and flags differences as suspect.
- Raises the cost of attacking a system by forcing an intruder to devise a different attack vector for each specific instance of a program.

**PLAN**

- Initial Prototype, Aug 07
- Add variable addressing diversity, Dec 07
- Expand the scope of diversity to system libraries, Nov 07
- Refined prototype, Feb 08
- Add application binary interface diversity, May 08
- Expand scope to the operating system, May 08
- Results from hardware support study, Jul 08
- Final prototype, Jul 08
**APPROACH**

• Track OS-level information flow provenance by assigning a unique identifier (color) to each potential malware entry point

• Color individual processes/data based on their interaction with potential entry points or other previously colored processes/data

• Color-based identification of malware contaminations

• Color-based reduction of log data to be analyzed

• Highlight event anomalies via abnormal color interactions present in logs

• Leverage virtual machine technology for tamper resistance of log coloring

**PLAN**

• Formal model of process (color) provenance for OS level flows, Jul 07

• Demonstrate a process coloring prototype in a malware scenario, Jul 08
  
  • Includes both server and client side solutions

• Evaluate the effects of color diffusion and mixing on malware warning and detection, including:
  
  • Profiling and visualization, Dec 08
  
  • Reducing false positives caused by legitimate color mixing, Mar 08
  
  • Tracking cross-border color mixing, Jul 08

• Deploy in a real-world environment, Feb 08 – Jul 08
100 Mb/sec for 100 Million Households: A Clean Slate Design for the Internet, S. Fraser, Fraser Research

Current Approach
- Security an ad-hoc overlay
- Mobility a late addition
- Dumb network, smart host

New Approach
- Security from the ground up
- Mobility an initial consideration
- Smart network, independent host

Privacy
- Access control for all conversations
- Authenticated endpoints

Robust secure service
- Rapid and automatic service restoration
- Infrastructure with multiple layers of defense

Universal mobility
- Unification of wired, wireless and satellite
- Every service, host and network can be mobile

Independent evolution
- A new host/network interface with signaling
- Extensible range of network styles and services

Incremental deployment
- Consistent with existing premises hardware
- Small initial change to host software
Ask the question: Starting with a clean-slate, how can we design Public and Private networks to be inherently secure from the ground up?

- Force the origin and intent of traffic to be explicit
- Private Networks:
  - SANE: All network connectivity governed by global policy; Implement secure namespace
  - Ethane: Ethernet-compatible prototype; Domain Controller and custom Ethernet switch.
- Public Networks: Work yet to start
- Technology Transition
  - Deploying at Stanford, UW-Madison and other research groups
  - Expect to work with commercial partners to transfer technology

Technical Results
- Ethane prototype deployed in research group at Stanford
- Software switch & Domain Controller implemented
- Hardware switch defined
Data Flow Analysis for Information Accountability and Security Enforcement, Jeremy Price, SWRI and Calvin Lin, UT Austin

UNCLASSIFIED

APPROACH

• Combine static analysis and runtime monitoring techniques to enhance the trustworthiness of applications
  - Detect vulnerabilities and automatically repair the affected application
  - Employ an annotation language to define security properties independent of any programming language
  - Statically identify potential violations in code
  - Insert runtime monitoring code into untrusted software

• New Start

PLAN

• Mid-term demonstration prototype kit, 9 months ARO
• Mid-Term Research Claims Evaluation Report, 9 months ARO
• Preliminary DDFA Design Document, 12 months ARO
• Final demonstration prototype kit, 18 months ARO
• Final DDFA Design Document, 18 months ARO
• Final report, 18 months ARO
Accountable Information Flow for Java-Based Web Applications / Andrew Myers, Cornell U.

Goal:
Information flow assurance for Java-based web applications

Motivation:
Explicit policies create new opportunities:
Users can have system enforce their own policies
Security policies exposed to users by automatically color-coding page content.
Compiler identifies code that can run securely on the client side as JavaScript -- easier and more secure

Approach:
Write web application code in Java with rich information flow annotations (Jif language)
Compiler checks information flows are secure
Some run-time checks can be necessary, overhead < 40%

Plan / Progress
New start fall 2007
Physical Unclonable Functions and Secure Processors, Srinivas Devadas, PUFCO Inc.

PLAN
• PUF enabled secure processor design and specification document (month 4)
• ASIC Implementation (month 12)
• Application Demonstration and Evaluation results (month 18)

APPROACH
• Build a secure processor based on inherent and unique chip delays to generate physically unclonable and tamper resistant secrets as a foundation for accountable information flow.
**Approach**

- Develop a next generation operating system infrastructure based on the concept of active attestation.
- Demonstrate that creation and use of a new generalized form of attribution certificates can reliably associate execution guarantees with specified software components.

**Plan**

- Complete stand alone infrastructure (3 months)
- Prototype active attestation of certification without assumptions (6 months)
- Implement introspection service (9 months)
- Build applications using active attestation (12 months)
- Implement an example network reference monitor (15 months)
**APPROACH**

- Develop novel logical techniques that capture both authorization and information flow in systems
- Show that these techniques can be integrated into systems to enforce these requirements for real applications
- Develop relevant policies
- Foundations for logics of affirmation and knowledge
- File system prototype

**PLAN**

- Specification of access control policies modeled after an Intelligence Community organization, Feb 08
- Formal logic, associated logic properties, and proof checker, Feb 08
- Demonstrate a proof-carrying authorization (PCA) prototype for creating and verifying proofs of access control in a selected file system, Jul 08

**Accountability for Information Flow via Explicit Formal Proof, F. Pfenning et. al., CMU / B. Witten, Symantec**

- fopen ("memo.txt")
- Prove: root says read
- pubkey_root: delegate (root, Alice, read("memo.txt"))
- Proof: ... root says ... delegate (root, Alice, read("memo.txt")) ...

Access policy:
root says read/write/delete/…

memo.txt
Trust Management Intrusion-tolerance, Accountability, and Reconstitution Architecture, (TIARA) H. Shrobe et. al., MIT / A. DeHon, Penn

APPRAOCH

• Provide for fine-grained tracking of data security context, non-bypassable enforcement of security policies, and application data provenance tracking.

• Use hardware based tag processing unit to support a broad range of access control and information flow models.

• Develop TIARA FPGA based hardware to ensure integrity of the memory structuring conventions and implement secure information flow. (Month 12)

• Develop TIARA software layers that enforce structuring constraints, access controls and data accountability (Month 12)

• Conduct active briefing book based demonstration. (Month 15)
Enforcing information flow policies via generation of monitors in Java (Card) Runtime Environments / A. Coglio, L. Errington, Kestrel Institute

Goal:
Provide high assurance of information flow policy enforcement in JavaCard
- current J(C)RE only provides “basic” protection against undesired information flows (e.g. no buffer overflows), but leaves the rest to applications

Approach:
synthesize extensions of J(C)RE that monitor application-level information flow policies to protect from Java application threats

Novelty:
- new capability (application-level information flow monitoring in J(C)RE)
- generative approach from formal specifications (vs. hand-written)

Plan / Progress
Capabilities developed along two dimensions
1st dimension: complexity of Runtime Environment
- JCRE (single-threaded)
- JCRE with multi-threading
- JRE for small devices (e.g. phones)
...

2nd dimension: complexity of information flow policies
- MILS (= Multiple Independent Levels of Security)
- MLS (= Multi-Level Security)
- more elaborate policies (e.g. expressed in temporal logic)
Provide Information flow guarantees via VMMs (e.g., Xen/shype) and applications (e.g., security typed languages)

**APPROACH**

- Develop practical information flow enforcement semantics
- Build application level enforcement of system information flow requirements
- Enforce and convey system information flow requirements
- Leverage virtual machine technology to simplify information flow integrity

**PLAN**

- Extend SELinux Labeled IPSEC mechanism to convey requisite security information (Month 8)
- Build Xen virtual infrastructure to enable verification of loaded software. (Month 14)
- Build a JIF based web browser that ensures secrecy and integrity of information flows (Month 15)
- Conduct functional tests and vulnerability analysis (Month 17)
End-to-End Semantic Accountability, D. Weitzner et.al., MIT / J. Feigenbaum, Yale / J. Hendler RPI

GOAL: Provide robust social and technical basis for trusting that Web-scale information systems are being used in accordance with the rules set out for them, where rules are expressed with reference to the semantics of the information under control.

APPRAOCH
- Access control through proof carrying authentication with access policies expressed over data semantics
- Transparent data usage logging for real-time compliance hints and a posteriori accountability
- Engineered as Web architecture components

PLAN
- Scenario descriptions, May 07
- Data purpose algebra specification, Jun 07
- Appliance design and initial prototype, Sep 07
- Prototype appliances:
  - Apache module, Nov 07
  - Client-side proxy, Dec 07
  - Accountability browser, Feb 08
- Final accountability prototype, Feb 08
- Final evaluation, Jun 08
Where do we go from here?

(How) can an innovative research program have real effects on national infrastructures?
Creating a National Cyber Defense Initiative

Highlights of Safe Computing Workshop (SCW) held Nov 2006 and subsequent activities
NCDI Key Contributors, Organizers and Catalyzers

James Gosler, Sandia
Cynthia Irvine, NPS
Keith Jarrin, NSA/NCSC
Carl Landwehr, DTO/IARPA
Karl Levitt, NSF
John Mallery, MIT CSAIL
Joe Markowitz, DSB
Bridget Rogers, Sandia
O. Sami Saydjari, CDA
Don Simard, NSA/NCSC
Adm. (ret.) Bill Studeman, DSB, NGC
Dick Schaeffer, NSA/IAD
Shannon Spires, Sandia
Alan Wade, DSB
William Worley, Secure64
People at SCW came from…

- Army Research Office
- BBN
- Boeing
- Carnegie Mellon U.
- Cryptography and Information Systems Surety
- Cyber Defense Agency
- Dartmouth College
- DePaul University
- Disruptive Technology Office
- IBM
- Institute for Defense Analysis
- Intel Corporation
- Kestrel Institute
- LynuxWorks
- MIT
- MITRE Corporation
- National Science Foundation
- National Security Agency
- Naval Postgraduate School
- Northeastern U.
- Northrop Grumman
- Oxford Systems
- Princeton University
- Sandia National Laboratories
- Secure64 Software
- SRI International
- Stanford University
- UC—Irvine
- UC—Santa Barbara
- U. of New Mexico
- U. of Penn.
- U. of Texas – Austin
Findings from SCW

Attackers Rule, Disasters are Likely
Short-term Measures Essential but Insufficient
Market Forces Will Not Change the Balance
Usability & Manageability Critical to Solution
New Technology Can Catalyze Major Changes

- Only a National Initiative Will Make a Real Difference
SCW Planning Phase

- Plan Via Selected Small Focused Working Groups
- Start with Four Working Groups
  - Program Group—Oversee and Spawn Groups
  - Vision Group—Create compelling vision
  - Threat Assessment Group—Case for Action
  - Architecture Group—Technical Approaches
- Resources Needed
  - Multiple Sponsors to Ensure Broad Support
- Profile: Eventually High Profile, but wait until have…
  - Vision, Plan, and Management Structure Worked Out
Vision: Over the next ten years transform the cyber-infrastructure to be resistant to attack so that critical national interests are protected from catastrophic damage and our society can confidently adopt new technological advances.

- **Transformation means we must learn how to build the new infrastructure and deploy it**
- **Learning how to execute this transformation will advance both technology and U.S. competitiveness in many ways**

Vision Working Group

Bridget Rogers, Sandia
George Cox, Intel
Tom Knight, MIT
David Mazieres, Stanford

Peter Neumann, SRI
Alan Wade, CIA (ret)
Grant Wagner, NSA
Elaborating the Vision

- "transform the cyber-infrastructure" refers not only to changes in technology, but to all types of change necessary to affect adequate protection (including laws, education, societal norms, software, hardware, and so on).

- "resistant to attack" does not mean "blocks all attacks" or "prevents all damage".

- "critical national interests" does not limit the vision to affect only those things owned or controlled by the US Federal Government.

- "protected from catastrophic damage" refers to situations which are so grave or so far reaching that a national response is required because the effects can't be remediated by individuals, communities, or corporations.

- "society can confidently adopt new technological advances" recognizes that an ability to rapidly adopt new technology is a vital aspect of national competitiveness.
NCDI Plan of Action

NCDI Scope
- Networks, hosts, warning, response, supporting technologies
- National effort from basic research to expedited delivery

Jump start (0-1 years)
- Begin work on gating science & technology immediately

Three parallel thrusts:
- Near term (1-3 years): Improvements
  - Shore up the existing infrastructure
  - Deliver proximate IA technologies for speedy deployment
- Medium term (3-5 years): Reengineering
  - Deploy technologies based on best IA engineering practices
  - Develop several domain-specific demonstration systems
- Long term (5-10 years): New platform and applications
  - Research foundational technologies
  - Create building blocks with agility in mind
  - Develop several priority applications based on new technology
  - Begin phased replacement of priority infrastructures

Start all three thrusts together after planning
Details need to be planned
Need Unorthodox Systems-Level Thinking

- Problem is highly complex, yet ...
- Integrated benefit required (not just local maxima)
  - Optimization of solution to one aspect of problem can degrade the whole resulting in
    - Duplication
    - Excessive complexity
    - Inefficiencies
    - Contradictions
- Need to create engineered building blocks designed to compose and be mutually supportive
- Must find ways to achieve synergistic design
  - In the face of malicious intent
- Consider radically different approaches
  - Example: non-von Neumann machines
Many areas to innovate

Security in System Development
Programming Languages
Computational Platforms
Data Stores
Networking
Operating Systems
Information Confinement
Operational monitoring / response
Scientific foundations
Teaching ourselves to build the infrastructure we want

- Claim: we have component technologies, special approaches, but not economical, full-up system development strategies that predictably yield systems both resistant to attack and capable of easy incremental extension
- Idea: Parallel system development competitions in different domains
- Goal: system(s) with sound assurance argument, capable of extension
- Domain examples:
  - Large scale IT infrastructure
  - Embedded control
  - Personal communication device
  - Election system
  - ....

Needed:
- System specification
- Evaluation Strategy
Your ideas go here
Thank You!

Questions?

Carl Landwehr
NICIAR Program Manager
301-226-9108
265-5188(s)
email: CLandwehr@casl.umd.edu
And finally, a current funding opportunity:

- A revision of the National Intelligence Community Enterprise Cyber Assurance Program (NICECAP) BAA (AFRL BAA-06-11-IFKA) has been released and is available at:
  
  http://tinyurl.com/yw7k9s

- A new focus area soliciting research on Privacy Protecting Technologies is added under the Accountable Information Flow thrust.

- White papers are due to AFRL
  
  >>November 2, 2007<<
NCDI-related Activities

- **DSB Net-centric Warfare Summer Study**
  - April-August, 2006

- **SCW Planning Meeting (DTO, MIT, NSA, NSF, Sandia, UPenn)**
  - August 21-22, 2006, MIT CSAIL, Cambridge

- **NSF-DTO-NSA Safe Computing Workshop**
  - (60+ top US IA experts from government, industry and academia looked at entire computing & networking stack)
  - November 29 – December 1, 2006, Sandia National Laboratory, Albuquerque
  - [https://og5.csail.mit.edu/scw/dist/](https://og5.csail.mit.edu/scw/dist/)

- **NSF CyberTrust PI Meeting (NCDI presentations)**
  - January 29-30, Atlanta

- **NSF-DTO-NSA Itanium STA Workshop**
  - (25 specialists examined Itanium as platform for STA and looked at programming language verification)
  - March 26-27 2006, MIT CSAIL, Cambridge
  - [https://og5.csail.mit.edu/cdi/itanium/](https://og5.csail.mit.edu/cdi/itanium/)

- **IA Leadership Workshop**
  - (20 USG IA leaders reviewed NCDI progress to date)
  - June 13, 2007, NGC, Reston
  - [https://og5.csail.mit.edu/cdi/ialw/](https://og5.csail.mit.edu/cdi/ialw/)