PANEL:
Cybersecurity Experimentation of the Future (CEF)

CSET Workshop

August 18, 2014

Goal of the Panel

- Engage the workshop participants in an interactive discussion of the experimentation capabilities and infrastructure needed to meet the challenges of tomorrow's cyber world
- The future will require a fundamental shift in the cybersecurity experimentation paradigm
  - Enable different domains to develop experimentation capabilities
  - Provide a means to unify and combine capabilities across domains
- This vision requires broad community input on future hard problems, infrastructure requirements, and capability needs

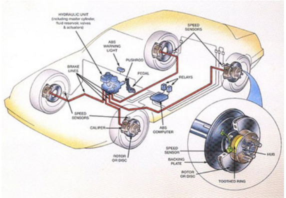
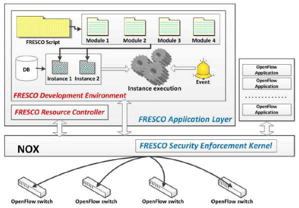
Motivation: Need Infrastructure to Support Scientific Experimentation

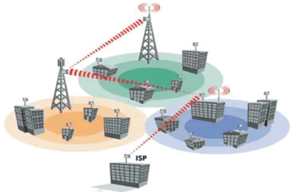
- Cyberspace is rapidly evolving with nearly every aspect of society moving toward pervasive computing and networking
- These changes bring real and wide-ranging cybersecurity threats and challenges that require new solutions based on sound scientific principles
- The scale and complexity of the challenges require that researchers employ experimentation infrastructure to enable discovery, validation, and ongoing analysis

Motivation: Experimentation Infrastructure Has to Keep Pace with Cyber Technology

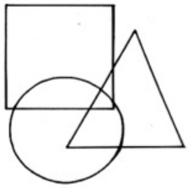
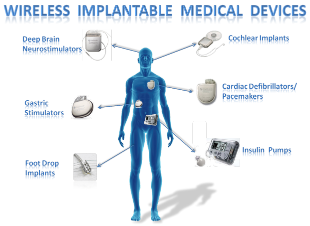
- Experimentation infrastructure is focused on today's needs, while anticipating and preparing for tomorrow's
- Current work extending existing infrastructure
  - Large-scale experimentation
  - Federated capabilities
  - Wireless
  - Software defined radio (SDR)
  - Etc.





- Need to move quickly to meet tomorrow's needs
  - Highly specialized cyber-physical systems (CPS)
  - Interdisciplinary experimentation
  - Modeling and reasoning about human behavior
  - Software defined networking (SDN)
  - Etc.

- Growing interest in a broad, accessible, and multi-organizational cybersecurity experimentation capability

A Definition of “Cybersecurity Experimentation Infrastructure”

- General purpose ranges and testbeds (physical and/or virtual)
- Specialized ranges and testbeds (physical and/or virtual)
- Software tools that supports one or more parts of the experiment life cycle, including, but not limited to:
  - Experiment design
  - Testbed provisioning software
  - Experiment control software
  - Testbed validation
  - Human and system activity emulators
  - Instrumentation – systems and humans
  - Data analysis
  - Testbed health and situational awareness
  - Experiment situational awareness
  - Other similarly relevant tools
- Specialized hardware tools – simulators, physical apparatus, etc.

Representative Cybersecurity Hard Problems

- Systems/software
  - Heterogeneity and scalability
  - Human element
  - Supply chain / root of trust
  - Increasing performance of security algorithms
- Networking
  - Software defined networking (SDN)
  - New network architectures
  - Privacy and anonymity
  - Trust infrastructure
  - Pervasive communications, w/o organizational and political boundaries
- Cyber physical systems
  - Embedded devices
  - Autonomous vehicles, smart transportation
  - Electric power, smart grid
  - Medical implants, body sensors, etc.

Where is Experimentation Applicable?

- Experimentation is about LEARNING

- Evaluation – not formal T&E
- To explore a hypothesis
- To characterize complex behavior
  - "Real" world
  - "What if" scenarios
  - Compare and contrast under different conditions
  - Trace a trend
- To complement a theory
- To understand a threat
- To probe / understand a technology

Questions for the Panel

- Identify key cybersecurity hard problems and future research that would be amenable to or benefit from experimentation
- What kinds of experiments need to be conducted in the future? What are some representative use cases? What experimental approaches and methodologies will best advance the research?
- Identify important characteristics, such as real world vs. emulated; centralized vs. distributed; independent vs. embedded; and fidelity, scalability, and repeatability
- What general capabilities (hardware, software, connectivity, etc.) are needed to support different types of experimentation, including specialized tools and domain-specific needs?
- What are the critical gaps between needed and current capabilities?

PANEL

- **Moderator:**
  - David Balenson, SRI International

- **Panelists:**
  - Stephen Schwab, USC Information Sciences Institute (ISI)
  - Eric Eide, University of Utah
  - Laura Tinnel, SRI International

DAVID BALENSON, SRI INTERNATIONAL

- David Balenson has over 25 years experience conducting and managing cyber security R&D projects for DARPA, DHS, and NSF. He participated in Phases I and II of the National Cyber Range program. Part of DHS team supporting the DETER Project. He is a co-PI for the NSF CEF study, which is a community-based effort to study current and expected cybersecurity experimentation infrastructure, and to produce a strategic plan and roadmap for developing infrastructure that supports tomorrow's research.

STEVE SCHWAB, USC INFORMATION SCIENCES INSTITUTE (ISI)

- Steve Schwab is a Project Leader at the USC Information Sciences Institute, leading a variety of security related research efforts. He has participated in a number of security experimentation efforts, including development of the DETER testbed and GENI testbed security architecture, as well as using DETER to conduct annual assessments of SAFER anonymity and anti-censorship technologies.

ERIC EIDE, UNIVERSITY OF UTAH

- Eric Eide is a Research Assistant Professor and Co-Director of the Flux Research Group at the University of Utah. The Flux Group is well known for inventing and operating public testbeds for networking, systems, and cybersecurity research, including Emulab (since 2000), ProtoGENI (since 2009), PhantomNet, and Apt (Adaptable Profile-Driven Testbed). Eric managed the Flux Group's participation in the National Cyber Range program, Phases I and II, and currently directs the Group's efforts under the DARPA CRASH program.

LAURA TINNEL, SRI INTERNATIONAL

- Laura Tinnel has been working in the cyber security research area since 1996, focusing on defensive systems design, risk analysis, and security experimentation. She serves as co-PI on NSF CEF study. Laura is in her second year serving as General Chair for the LASER Workshop, which is focused on scientific experimentation in cyber security.

Stephen Schwab
USC Information Sciences Institute
August 18, 2014

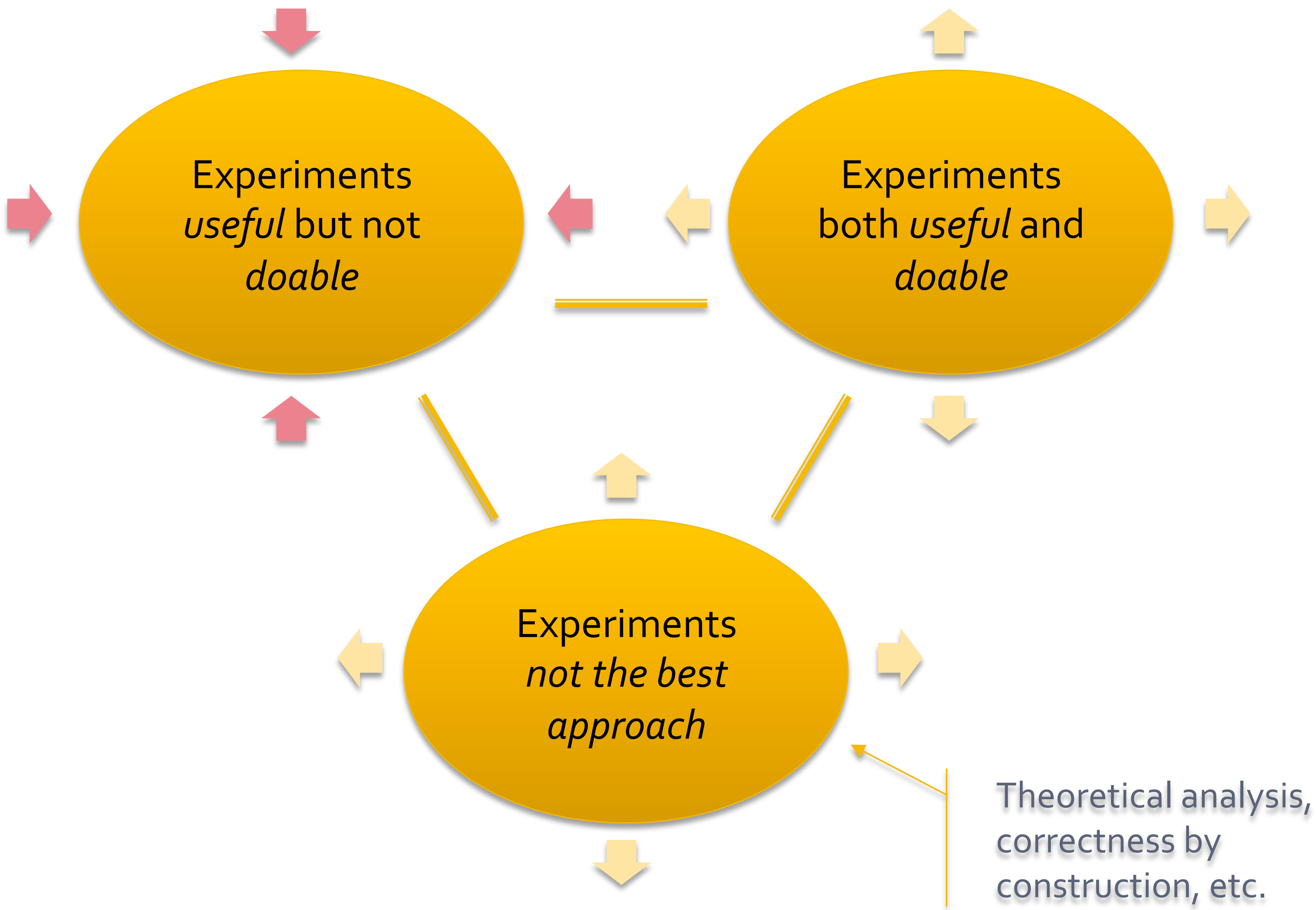
Future Cybersecurity Experimentation
Thoughts on
Hard Problems and Capability Needs

Thoughts on Cybersecurity Hard Problems

- **Problems that…**
- Yield Insight
- Frame Further Advances
- Represent Grand Challenges

- Thought: We should seek hard problems that foster a "big science framework", and serve to roughly structure the work of many individuals.

What are levers to shift feasibility frontier that drives fundamentals of Cybersecurity Experimentation ?



Characteristics of Cybersecurity Experimentation

- Cyber Security (and hence Cyber Security experimental work) is *intrinsically hard:*
  - *Large, complex, decentralized* systems
  - Focused on *worst case behaviors and rare events*
  - Intrinsically *multi-party* and frequently *competitive* scenarios

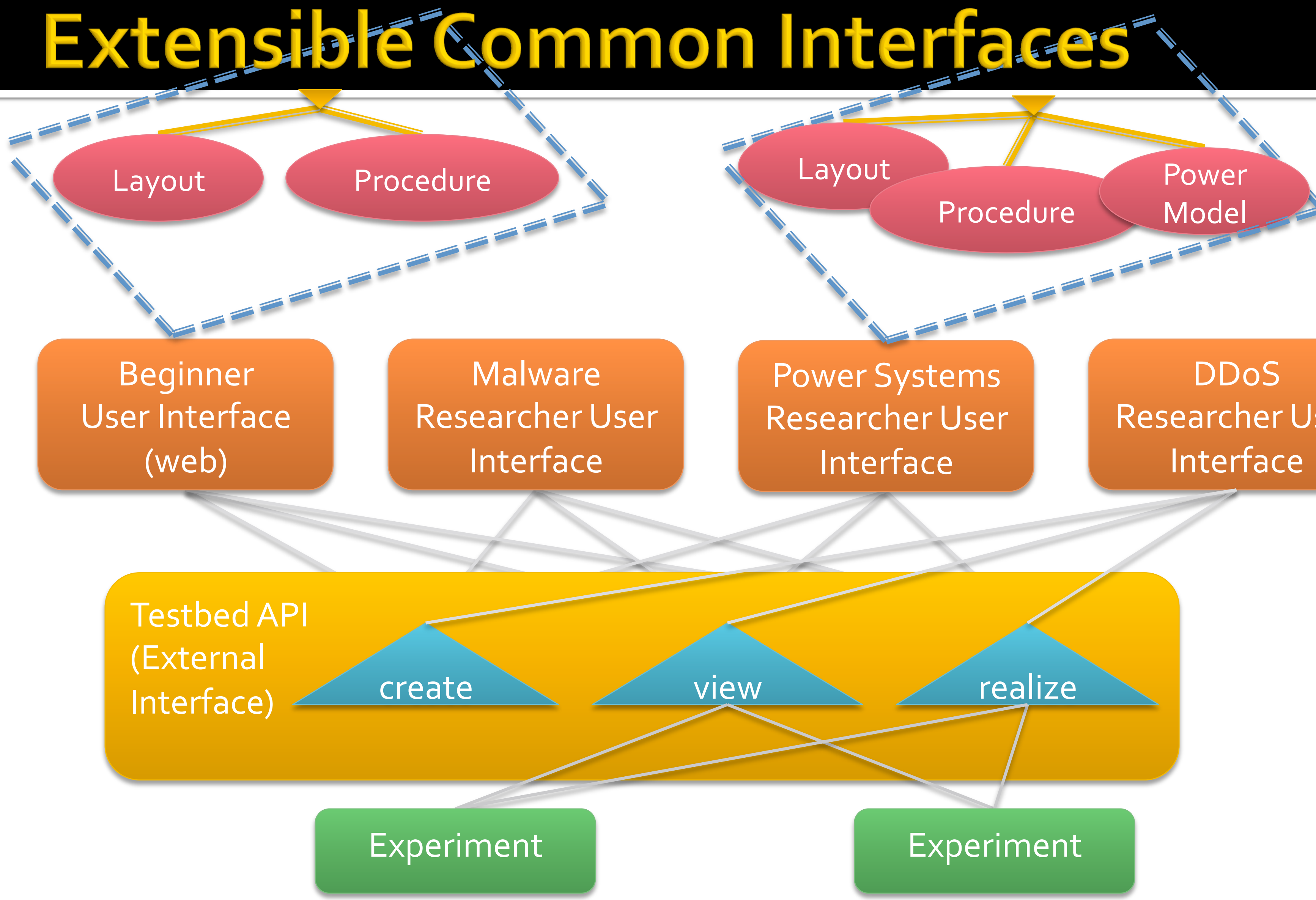
- Experiments and scenarios that are not sufficiently
  - Well-framed
  - Scaled
  - Realistic
  - Etc.

  to be *valid* are instead by definition *misleading*

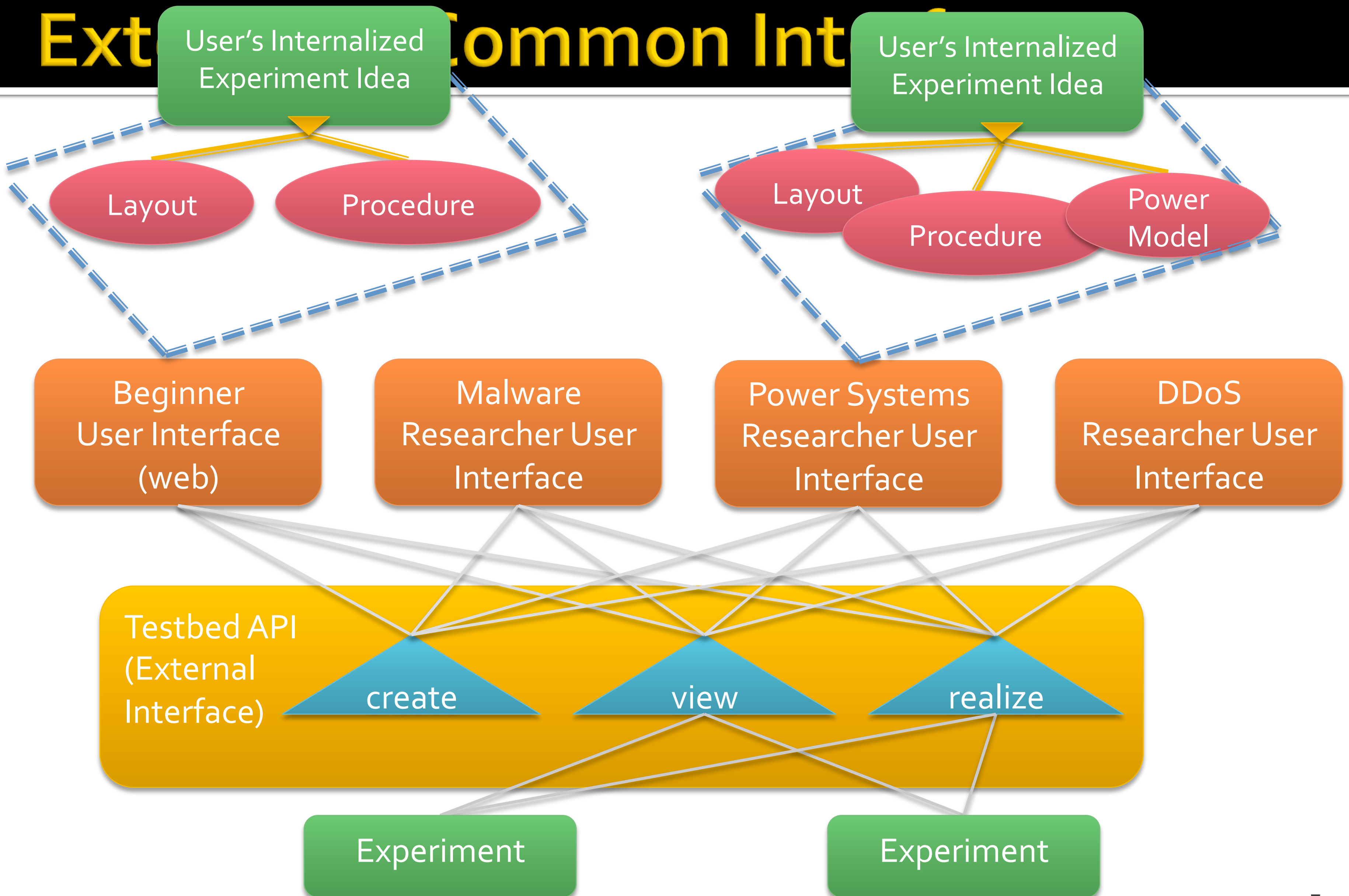
Thoughts on Capability Needs

- An architectural view of
  experiment management capabilities –
- Motivated by the need for experiment management to advance in parallel with theory/rigor and testbed capabilities
- Key Observation: view tools as maps from user-centric to testbed-centric views

Architecture: Extensible Common Interfaces



Architecture:
Ext Common Int



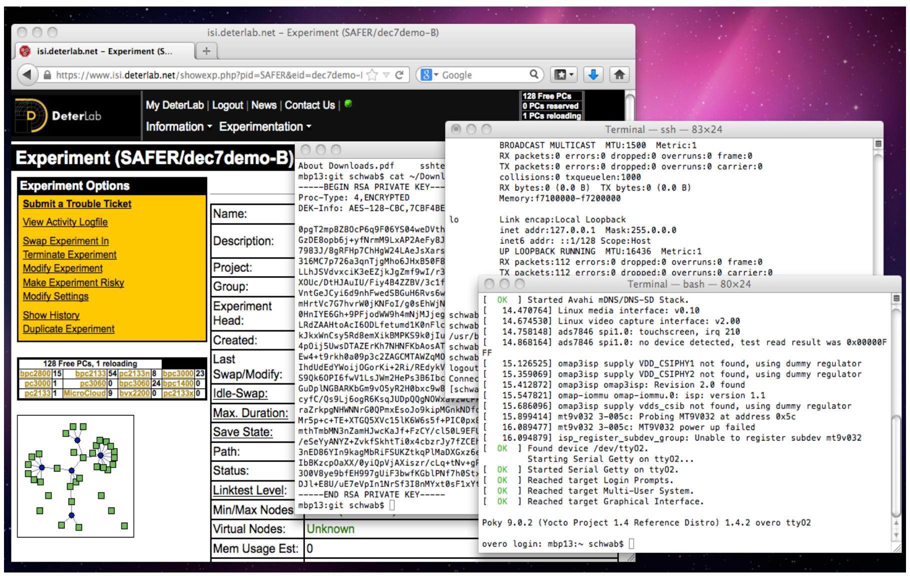
Architecture:
Aspects Of Experiments

- Universal Aspects
  - Layout
    - How to Configure Resources
  - Procedure
    - How to Manipulate Resources
  - Data Collection
    - What Data to Gather for Successful Experiment
    - How to Gather Data without Disruption
- Relationships between Aspects
  - Constraints involving one or more aspects
  - Establish Conditions Necessary for Valid Results
  - Ensure Limits on Interactions with Internet or Other Networks
  - …
- Specialization of Aspects
  - Domain Specific
    - Financial SLA Models, Traffic Characteristics, …
  - User Interaction (Purpose of Experiment)
    - Exploratory vs. Demonstration vs. Rigorous Hypothesis Test
  - …

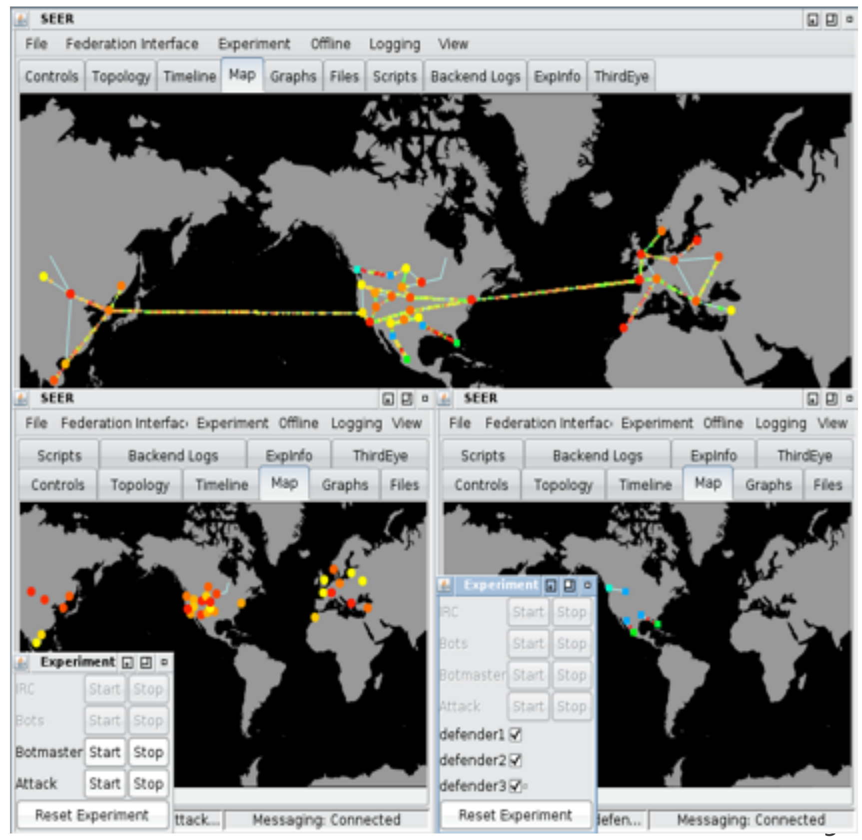
Aspect changes reflect user-centric to testbed-centric view

Architecture:
User Experience / Customizability

Power Users:
Command Line shells,
Low-level APIs.
Exposes full power of the testbed



Domain Specific Users:
User-Interfaces tailored to the skills, background, and needs of a community.
Exposes user-centric aspects of experiment



Tools provide the *mapping* between:
user-centric abstractions
and
testbed API abstractions

Architecture:
User Experience / Customizability

- Thought: the “right” interfaces enable customized tools
  - Present a User Experience tailored to
    - The domain of experimentation
    - The skills and background of the user community
    - The terminology, workflow and methods traditionally preferred within an established community-of-interest

Summary

- Hard problems must both fit within and shape an overarching framework for cybersecurity research
- Requisite ``experiment management capabilities" align with an architectural view
- Motivated by the need for experiment management to advance in parallel with theory/rigor and testbed capabilities
- Key Observation: view tools as maps from user-centric to testbed-centric views

CSET ’14 CEF Panel: Position Statement

Eric Eide
University of Utah
eeide@cs.utah.edu

What are we good at?

- providing testbed and range “iron”
- configuring those resources

- we are getting better about sharing, enabling repetition, and enabling reproduction

What are we not good at?

- *experiment design*

Experiment design

- being precise about what we want to measure
  – response variables
- being precise about what we want to achieve
  – exploration: determining what factors matter
  – optimization: of response variable
  – comparison: new method versus old method
  – stability: of response variable
  – …

Experiment design

- identifying factors
  - controllable factors
  - uncontrollable factors
  - hidden factors
- e.g., lots of work shows that there are many hidden factors in systems performance research

Performing experiments

- dealing with
  - experimental error
  - measurement error
- repetition
  - performing repeated measurements
  - repeating experiments
- randomization
  - randomized trials
  - blocking

Performing experiments

- factorial experiments
  - versus one factor at a time
  - choosing how to set the factors

“Rampant realism”

- a consequence of our ignorance of experiment design: *“rampant realism”*

- “I don't really know what matters, so I will simply require that everything be real”

- real machines, networks, OSes, applications, actors, on-disk data, live Internet, malware, …

But we *require* realism, don't we?

- malware only works under real conditions
  - it requires real things, like Windows and Android and particular applications
  - it wants to talk to outside (uncontrolled) entities
  - it breaks through abstractions, so our test harnesses must be real "all the way down"

- impact: only solutions that work in real environments matter
  - "different" is not necessarily better

Realism can have a high cost

- time/effort/financial cost of setup
  - software, hardware, time to configure everything
- difficulty of varying the setup
  - increases time to run multiple trials, etc.
- difficulty of obtaining measurements
- difficulty in scaling up
- difficulty in setting up far future/far past scenarios

CEF position: experiment design

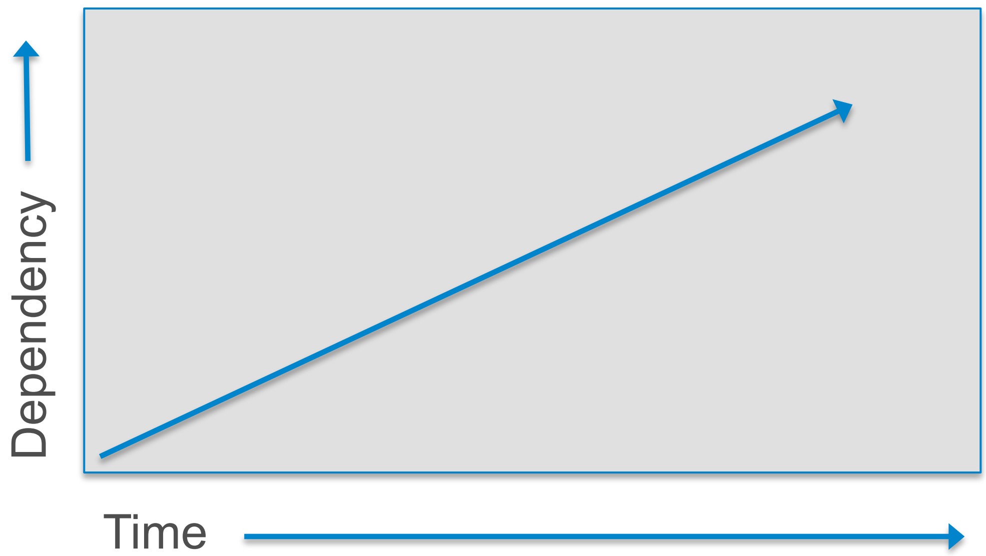
- more education about experiment design and analysis
- better support for people to design effective experiments
- methods for overcoming the problems caused by "rampant realism"
  - identifying what actually matters

Thoughts on Cybersecurity Experimentation

Going where no researcher has gone before

Laura S. Tinnel
SRI International

We Have a BIG Problem



Security not designed into new cyber technologies

Researchers cannot keep up

Significantly More vulnerable

State of the Art in Cyber Experimentation

- Existing testbeds
  - Generally available: mostly general purpose for IT systems and network testing
  - Proprietary: specialized to domain
- Most researchers stand up their own test environment
  - Existing testbeds are inaccessible or don't meet unique requirements
  - Often need to write software to control experiment, collect data
  - Pros: Ability to tailor and control, don't need to share resources
  - Cons:
    - Valuable time that could be spent on research is spent building test infrastructure
    - Error prone due to all the moving parts created by humans
    - Not as easily sharable to allow peer-based examination and replication of results

Enable researchers to do what they do faster, better

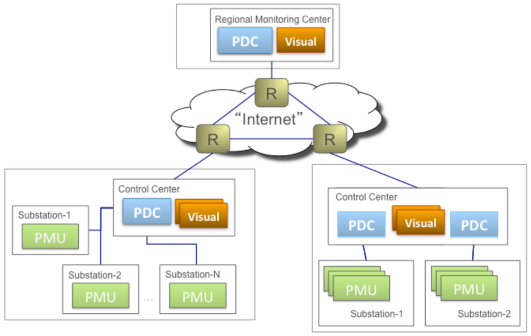
Incremental Improvement on Today's Process

- Shared community repository of testbed research infrastructure
  - General experiment framework with plug/play architecture
  - Shared domain-specific topologies, models, etc
  - Accessible to all researchers
    - Publically available, e.g., via GIT
    - Open source

Improve the Science, Enable Fast Iterations

- Software development analogy: Integrated Experiment Environment (IEE)

- Encode in Experiment Specification Language, specify
  - Nodes, topology, actors
  - Resource requirements
  - Metrics plus data to be collected to validate hypothesis
  - Policies – restricted access, checkpoints, pause/resume execution constraints



Improve the Science, Enable Fast Iterations

- Compile
  - Parsers / syntax checkers
  - Lint-like validation
  - Optimization – predictive of resources needed, recommend alternatives for better resource use

```
Warning: missing baseline.
```

- Link
  - Static – configure topology statically & use snapshot images
  - Dynamic – build images dynamically
  - Models / libraries

```
Using: star-topology-21
Using: microsoft-windows-10-base
Building: linux-apache-server-generic
```

- Load
  - Dynamic federation of testbeds (w/ provisioning requirements/policy)
  - Provisioning resource

```
Allocating: generic-node 1
Loading: node-1 microsoft-windows-10-base
Connecting: iowa-state-cps-testbed
```

Improve the Science, Enable Fast Iterations

- Infrastructure validation / pre-check
  - Specified node, service, network configuration

- Execution / run-time
  - Configuration
  - Monitoring
  - Termination
  - Restart to prior checkpoint

```
Fault: expected service HTTP not responding.
Checkpointing enabled.
Reset to checkpoint? (y/n)
```

- Step Debugger
  - Assistance to determine why an experiment fails to execute as expected

Improve the Science, Enable Fast Iterations

- Post Analysis
  - Validation of execution
  - Semi-automated knowledge extraction

- Shared community library of experiment designs, models, data & knowledge gleaned
  - GIT hub like
  - Building blocks: primitives, libraries, descriptions
  - Knowing / tagging provenance (content)
  - Knowledge dependency graphs

Laura S. Tinnel
laura.tinnel@sri.com