Electronic Preprint for Journal of Information Processing Vol.20 No.4

Invited Paper

The DETER Project: Towards Structural Advances in Experimental Cybersecurity Research and Evaluation

Terry Benzel[1,a)] John Wroclawski[1,b)]

Received: March 2, 2012, Accepted: June 18, 2012

Abstract: It is widely argued that today's largely reactive, "respond and patch" approach to securing cyber systems must yield to a new, more rigorous, more proactive methodology. Achieving this transformation is a difficult challenge. Building on insights into requirements for cyber science and on experience gained through 8 years of operation, the DETER project is addressing one facet of this problem: the development of transformative advances in methodology and facilities for experimental cybersecurity research and system evaluation. These advances in experiment design and research methodology are yielding progressive improvements not only in experiment scale, complexity, diversity, and repeatability, but also in the ability of researchers to leverage prior experimental efforts of others within the community. We describe in this paper the trajectory of the DETER project towards a new experimental science and a transformed facility for cyber-security research development and evaluation.

Keywords: cyber-security, testbed, experimental research

1. Introduction: Challenge and Response

Any strategy for constructing and deploying robust and secure cyber-systems will span the full lifecycle of the deployed system. The system is designed and built as robustly as possible within technical and business constraints, deployed, and then continually patched and updated in response to newly discovered design flaws, bugs, and emerging threats.

Today's implementation of this strategy is often perceived as primarily reactive. That is, a great deal of emphasis is placed on response to empirically discovered weaknesses and threats through patches and updates to already-deployed systems and components.

There is good reason for this. The alternative is *extremely hard*. Truly effective "designed in" security and robustness requires system engineers to simultaneously address several challenging dimensions, including:

- Technical properties of the system, its components, and its environment, including the system's behavior under stress and in rare, poorly understood corner cases.
- Human factors, the system's usability, and its sensitivity to human behaviors and judgment.
- Business and economic factors, such as cost, time to market, and inability to quantitatively value security.

Despite these difficulties, the need to move from the reactive model to a more proactive, design-driven security and robustness model is reaching critical levels. Several factors contribute to this need. Among these are:

- The increasing centrality of cyber systems to everyday life – ranging from the social Internet to complex networked cyber-physical systems that are increasingly elements of national critical infrastructure.
- The increasing complexity and interdependence of these systems, including legacy systems and other systems that are slow to evolve;
- The significant practical difficulty of managing a timely, controlled, cost-effective patch and update process across thousands or millions of system elements.
- The disconcerting reality that it is much easier to attack than to defend a cyber system, because the problem is inherently asymmetrical. Malicious attackers can rapidly evolve and easily proliferate tools and attacks, and frequently can deploy such attacks from anywhere, using unknown weapons, towards targets that are of necessity well known, with cyber defenses that must be known, effective, and affordable for the targets' operators.

Considering these factors, our long-term goal must be to increase cyber-security by decreasing fundamental vulnerability – to build target systems that are less vulnerable to begin with. Accomplishing this objective will likely require advances in many dimensions, ranging from theoretical and formal methods to trusted electronics to the psychology of user interfaces. The mission of the DETER [1] Project is to develop and make available to the larger cybersecurity research community key advances in one

[1] USC Information Sciences Institute, CA 90292, USA
[a)] tbenzel@isi.edu
[b)] jtw@isi.edu

This paper draws significantly from an earlier work titled "The Science of Cyber Security Experimentation: The DETER Project" [26].

This research was sponsored by the US Department of Homeland Security and the Space and Naval Warfare Systems Center, San Diego (Contract Number N66001-10-C-2018), and by the US National Science Foundation (Contract Number CNS-0831491). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Department of Homeland Security, the Space and Naval Warfare Systems Center, San Diego, or the National Science Foundation.

such area: that of methodologies, technologies, and facilities for rigorous and effective experimental cybersecurity research and evaluation.

1.1 DETER's Contribution

Two key challenges to the advance of any engineering field are gaining deeper *understanding* of the field's domain, and carrying out accurate *evaluation* of the eventually engineered systems, technologies, and components. In each of these cases, experimental research, exploration, and evaluation has time and again proved crucial.

In the cybersecurity domain, the need for such advance is widely recognized, as are severe limitations in current practice. A recent panel of respected experts [25] describes a "typical" security experimentation process of the previous decade as:

- Have an idea for a new tool that would help to address a specific threat;
- Develop the software and assemble to tool – the majority of the effort expended;
- Put the tool on a local network, and attack a target system protected by the tool;
- Show that the tool repels the attack, and write up the "results";
- Any or all of: publish a paper, open-source the tool, start up a company.

This caricature serves as a foil for the same panel to describe a science-based approach:

- Define a model of real-world large-scale computing systems that need more robust systemic security;
- Create an experimental apparatus with a design that matches the model, including relevant parts of the scale and complexity of the real system;
- Perform experimental research that demonstrates the modeled scale and complexity;
- From experimental observation and data, extract understanding of potential improvements in strength and mitigation of weakness of the modeled systems;
- Enable others to share and leverage experimental results and insights – and vice versa – via reproducible artifacts and methods of repeatable experimentation.

While space precludes a full description in this paper, the agenda of the DETER project addresses each of these elements. The project's overall objective is to enable and foster within the cybersecurity research and practice community a science based experimental research approach applicable to complex cyber and cyber-physical networked systems and components. To meet its objective the project has three elements:

- A research program, to advance capabilities for cybersecurity research and experimental methodology;
- An operational, shared experimental facility, to gain a broad base of users and experiments, and to support technology transfer of our and others' research;
- A community-building activity, to support collaborative science that is speeded by effective, efficient leveraging of experimental results and knowledge.

In this paper we first outline the project's history, which motivates its current agenda. We then focus on key elements of the research program, starting with 4 key observations from researcher experience and then discussing current research undertaken in response to these observations.

2. Motivation and History of the DETER Project

The DETER project's creation grew out of three related observations made in the early 2000's within the computer and network security research community, funding organizations, and security product companies. The first of these observations has been discussed in the introduction to this paper:

- Security technology and development was largely re-active in nature.
- Security technology development was slower in pace than the evolution of existing threats and the emergence of new threats.

A second observation was that the nature of the problem was fundamentally changing with the growth of the Internet:

- Current generation (i.e., circa 2000) widely deployed security technology (host security, communication security, network boundary control) could be tested with common equipment at small scale.
- Emerging threats, not addressed by deployed security technology, operate at Internet scale (worms, DDOS); requiring radically new classes of defense, and hence radically new evaluation strategies for these defenses, that focus on scale and aggregate behavior.
- New security approaches (e.g., behavioral anomaly analysis, data mining) also need large scale and highly varied testing.

A final observation related to the nature of the industry and the community:

- Security innovators lack the facilities to test new security technology in test environments with scale and fidelity to the real deployment environment, and typically construct their own test environment with little or no leverage from the testing work of other innovators.

A consequence of these observations was that promising new security technologies, often from innovators with limited testing resources, fared poorly when tested by applied security practitioners in real deployment environments [3]. In such cases, technology transfer was problematic because of significantly lower effectiveness outside the innovator's limited experience and test facility. Yet, in many cases, commercial organizations did not find it cost effective to engage in further development to increase effectiveness.

With this background in 2001–2002, one of the several factors of cyber-security deficiency seemed to be addressable: the lack of publicly available testing facilities with significantly greater resources and flexibility than the limited test environments of most innovators, and greater fidelity to real deployment environments. A US DARPA-sponsored report [4] called for and stated requirements for a national cyber-defense technology test facility. One result of that report was the impetus for the US National Science Foundation (NSF) and the recently formed US Department of Homeland Security (DHS) to define and fund the project that was the first phase of DETER.

The initial focus of DETER was to build such a national testbed, enabling cyber security innovators to test new technology at larger scale than could be assembled in most individual laboratories, with more complex test fixtures, and designed to be more representative of real deployment environments. The first-phase DETER project (led by USC/ISI, UC Berkeley, and Sparta, Inc.) was funded by NSF, leading to the assembly of the network and physical resources, development of controls and user interfaces for experimenters, assessment and integration of existing tools, and the creation of a collaborative community of researchers.

The testbed became operational in March 2004. The first DETER Community Workshop was held in November 2004, with working groups of researchers who published refereed publications on work performed in the DETER testbed covering, e.g., DDOS defense [5], worm dynamics [6], worm defense [7], and detection of routing infrastructure attacks [8]. The ensuing years saw maturation of the testbed through use and expansion, and growth of the research community with a greatly increased breadth of activity.

A natural, and desired, result of this activity was that researchers and community collaborators began to study and improve the technology of the testbed itself, in such areas as experiment automation and construction [10], benchmarking, scaling via hypervisor usage, malware containment, and federation [9], all now central components of DETER technology.

In the second phase of the project, 2007–9, the results of this “research on research” – our exploration of novel technologies and methodologies for cyber-security research – were put into practice in the testbed, which was also expanded in capacity. The result was the evolution from the DETER testbed to DeterLab, a shared virtual laboratory composed of three elements: the underlying testbed facility and hardware resources, technology for using and managing the resources as test fixtures, and a growing variety of tools and services for experiment support.

With the technological maturity achieved in this phase, and the experience gained from supporting over 1,000 researcher team members, the stage was set for a third phase of DETER project activities that focus increasingly on research and development in the areas of cyber-security experimentation methodology, infrastructure, and tools, with the aim of creating new experimental capabilities and approaches that directly target the challenges of Section 1.1.

The balance of this paper focuses on these new capabilities and approaches. We outline four research challenges identified by observing DETER usage over time, and discuss elements of the current research program that respond to each of these challenges. We describe some additional research activities briefly, and conclude by discussing how the integration of these activities advances DETER’s goal to serve as a unique, advanced scientific instrument for experimental cybersecurity research.

3. Observations from the DETER Experience

Here we describe several observations from ongoing user experience with DETER that have guided our ongoing research. This guidance derives from observing norms of researcher activity that emerged in the 1st and 2nd phases of DeterLab use, as well as from other networking research testbeds [14], [15], [16], [17] and our own study of the experimental research process. Each of the points described here motivates an ongoing research activity within the DETER project.

3.1 The Need for Flexible Scale and Fidelity

Two important contributors to the validity of any experimental scenario are the scale of the scenario and the accuracy, or fidelity, with which relevant features of the real world are captured and modeled in the experiment.

The initial DETER testbed, drawing on its Emulab [11] roots, implicitly assumed a single design point, in which individual testbed “nodes,” implemented by general purpose PCs, modeled individual nodes in an experimental scenario. This assumption led to relatively small size for the largest possible experiment, based on the number of PCs in the DETER facility, and to a single, fixed level of modeling fidelity, dependant on how accurately a PC could emulate the particular network element being modeled.

Of course, this is not the only approach. Though the technology was not well developed at the start of the DETER project, virtual machines (VMs) and other resource sharing constructs can be used to support larger scale experiments at the cost of some fidelity. However, simple use of VMs rather than hardware nodes would simply lead to a different, but still fixed, design point.

As we built out DeterLab, not only did we want to increase the scale of experiments, but we also recognized that many of our users’ experiments did not require any single fixed fidelity, but rather required *different* degrees of fidelity in different parts of the experimental scenario. We recognized a class of “multi-resolution” experiments [21] in which:

- Some parts of an apparatus require high-resolution nodes with high fidelity;
- Some other parts require a lower degree of resolution and can represent real computing at a larger scale;
- There is a “scale of scaling” with points that range from high fidelity and linear scaling, to low fidelity and high scalability;
- Different points on the scale will be enabled by different mechanisms for emulation and simulation.

As a result of this observation, we began to explore methods to incorporate into a single experimental scenario multiple representation methods that together provide a full *spectrum* of scale-fidelity tradeoffs for experimental system components. The following is a partial list of examples:

- A single hardware node running a single experiment node, either natively, or via a conventional Virtual Machine Manager (VMM) supporting a single guest OS;
- A single hardware node running several virtualized experiment nodes, each a full-blown conventional Virtual Machine (VM) on a conventional VMM;
- A single node running a large number of lightweight VMs on a VMM designed for scaling the number of experiment-nodes with limited functionality;
- Representation of individual experiment nodes as threads of

execution in a large-scale thread management environment;
- Large-scale software-based network simulation [22].

Further, we recognized that these methods would be more useful to experimenters if all methods were part of a single unified framework for the construction of composable experiment scenarios. Essentially, such a framework would allow the computational and communication resources to be allocated in the *most effective way* to support the scale and accuracy required by a particular experiment, without the assumption of any predefined mapping between testbed resources and experiment scenarios.

Our approach to such a framework is to base on it on an abstract fundamental building block called a "container." A container represents experimental elements at the same level of abstraction, and is the basic unit of composition for constructing an experimental scenario. The container-based methodology is a key part of pursuing some important goals:

- Leverage DeterLab's physical resources more flexibly to create larger scale experiments;
- Enable experimenters to model complex systems efficiently, with high resolution and fidelity for the things that matter most to the particular experiment and increased abstraction for the less important elements;
- Reduce the experimenter's workload of experiment apparatus construction, enabling larger scale apparatus with lower levels of effort.

3.2 The Limits of Experiment Isolation

Initially, the intent of the DETER design was that experiments would proceed in complete isolation from each other and the external world. For example, the intended methodology for malware experimentation in DETER was to observe and capture malware in the wild, and then to run the captured malware in a simulated network in the testbed, fully isolated from the public network by a number of extremely rigid segregation measures [18].

This approach quickly proved limiting. Much software (both desired and malware) has non-functional or non-deterministic behavior in this scenario: for a number of reasons the behavior of a copy in the testbed may have low fidelity to behavior in the wild. Another limitation is a timing issue: for emerging threats, the time required for accurate capture from the wild may introduce delays in the experimenter's ability to test.

As a result, we began to explore a methodology for "controlled Internet access" from DeterLab, in order to explicitly support and control valuable and effective, but also potentially *risky*, experiments – that is, experiments that pose a risk to the outside world, or are at risk from the outside, in addition to the inherent risk of using malware in a testbed or test lab. Some examples of experimentation to be enabled by risky experiment management:

- Place in DeterLab some targets for malware in the wild, and observe in a controlled environment the methods of attack; more expedient than trying to capture the malware and accurately replicate its execution in the test environment. Researchers at CMU and UC Berkeley were some of the first to use the new controlled internet access in order to attract drive-by downloads. The scenario was: a node in DETER visits some Web page, gets infected by malware and that malware instructs it to go visit other Web pages in unpredictable manner. Then they were able to use the infected nodes and behavior to analyze the malware [19].
- Place in DeterLab some peer computing elements to join in collaborative computing in the wild, for example real anonymity services and infrastructure, the operation of which is dependent on small-time changes in behavior that are non-deterministic; more expedient than replicating a privacy network at scale in a lab, and have the simulated behavior have high fidelity to real-world behavior.
- Place in DeterLab some nodes to serve as bots in botnets, to observe bot/botmaster behavior; more expedient than trying to replicate botmaster behavior with the same software and the same human operator behavior as real botmasters.

The common theme – whether or not malware is used in DeterLab – is that many experiments of interest depend on some level of interaction with the external world. Partly in response to experimenter requests, and partly from our desire to expand DeterLab's capabilities to accommodate this common theme, we began work on a structured approach to flexibly manage this sort of interactions.

Our work builds on a single, simple fundamental observation:

- If the behavior of an experiment is completely unconstrained, the behavior of the host testbed must be completely constraining, because it can assume nothing about the experiment.
- However, if the behavior of the experiment is constrained in some known and well-chosen way or ways, the behavior of the testbed can be less constraining, because the combination of experiment and testbed constraints together can provide the required overall assurance of good behavior.

We refer to this approach as Risky Experiment Management (REM) T1-T2 because it combines two sets of constraints, derived from the above observation, to limit the overall risk of the experiment. We call the first sort of constraints "experiment constraints" or "T1 constraints"; these are constraints naturally exhibited or explicitly imposed on the experiment. We call the second class of constraints "testbed constraints" or "T2 constraints"; these are constraints imposed by the testbed itself. We often refer to overall concept as the "T1/T2 model."

Implementation of the REM-T1/T2 approach [20] will require tools for formal definition of the experimenter's requirements – defining the T1 transformation – and methods and automation for defining the additional constraints that define the T2 transformation. These advances will be required for risky experiments to be defined, controlled, and permitted with more assurance than experiment-specific tunnel nodes. Section 4.4 provides further information on future efforts on REM-T1-T2.

3.3 Advancing Experiment Construction

DETER's initial tools for construction of experimental apparatus were inherited from Emula [11], the base technology of the original DETER testbed. These tools provided sophisticated but low-level capabilities for managing the physical computing and network resources of the testbed, to create emulated networks within which an experimenter's activity took place.

For highly skilled researchers, this toolset was useful, because it provided a mode of operation in which every detail of a test network could be specified. However, we quickly confirmed that the "expert mode only" approach was limiting for many of our researchers, some of whom were less concerned with network-centric security research, and more oriented toward security research that did not depend critically on an exactly specified network environment.

- Novice DeterLab experimenters with modest research experience faced a steep curve to learn how to create an emulated network of low complexity, but useful for testing.
- For very experienced cybersecurity researchers starting work in DeterLab, there was also a steep curve to learn how to create an emulated network of moderate complexity and realism sufficient for their work.
- Even researchers entirely capable of using the expert-mode tools could gain increased efficiency from other, higher level approaches.

The root limitation of this toolset derived from two separate properties. First, the only vehicle available to define experiments required writing a detailed description file for the desired experiment scenario. Second, description of a new experiment required starting from scratch.

In other words, the experiment definition methodology lacked abstraction and re-use. Acceleration of the pace of cyber-security research was blocked by the necessity of each experimenter needing to specify a great deal of structure, much of which was not critical to their needs, and without recourse to others' work.

Our lesson was that the construction part of the experiment lifecycle needed considerable additional automation, new methodology, and supporting features for abstraction, data hiding, and re-use. As our research on higher-level experimental infrastructure support turned to Experiment Lifecycle Management (ELM) – a concern for tools and methodologies that would assist in the full lifecycle of an experiment, from initial conception through design, execution, analysis, data collection and eventual archiving. We further incorporated a focus on sharing between researchers at each of these stages, adding the objective that a new experiment should be able to "stand on the shoulders of previous experiments, rather than standing on their feet."

3.4 From Data to Knowledge

An inevitable result of deploying improved experiment construction and execution technologies to DETER users was that DETER experiments quickly became more realistic, more data-intensive, and dramatically larger in scale.

This success quickly brought into focus a growing need for DeterLab experimenters: sophisticated tools to analyze, understand, and learn from experimental results. As DeterLab facilities have matured with scale and power and data capture capability, and as observation of the behavior of a running experiment drove improvements in data collection, the result was, for many experiments, orders of magnitude more output data to be analyzed from each experiment run.

Further, not only the size, but also the structure and complexity, of the datasets increased. In addition to log analysis tools to help deal with raw data size, there was a need for other methods – and automated support for them – to analyze data in at high level, in terms of the intended semantics of the experiment run, and ultimately to proceed from data analysis to actual experimental science: proving or disproving a hypothesis, or stating knowledge of malware behavior, or use of metrics for effectiveness of countermeasures.

In other words, experimenters need both tools and methodologies for transforming experimental data into results and knowledge. This lesson learned served to underscore the importance of our research work on narrowing this large "semantic gap" as part of our research efforts on Experiment Lifecycle Management.

4. Current DETER Research: a Snapshot

Our current research program includes, but is not limited to, activities related to the observations and conclusions described above. In this section, we outline one portion of the program, focusing on the ability to create experiments using high level design tools and models and then to implement and execute in DETER experimental scenarios with hundreds of thousands of elements and necessary realism and fidelity. Together these capabilities give the experimental cybersecurity researcher dramatically increased capabilities over today's norm.

4.1 Experiment Lifecycle Management and Montage

Experiment Lifecycle Management (ELM) is our name for the set of tools and processes that help the researcher to manage experiments through a full scientific lifecycle from conception to final archiving. As they develop, the ELM tools will become the primary interface between DeterLab and its users. As such, our focus is on both technical capabilities and strong user interface and human factors design.

ELM is an outgrowth of work on our first generation GUI experiment support workbench, SEER [10]. Indeed, many of SEER's capabilities, including experiment monitoring and visualization, are carried over into the next generation workbench, the Montage Workbench. However, the Montage workbench goes well beyond SEER in its capabilities and concepts.

One critical aspect of Montage focuses on the general concept of understanding the full range of objects that an experimenter uses. DeterLab has grown to include a large number and variety of objects available to experiments. With that growth has come the challenges of giving experimenters the tools need to effectively manage their working set, and (critically) to effectively share with other experimenters.

The objects used by an experimenter include scientific, physical, communication, and computational resources used in an experiment. Also included are models, designs, procedures, programs, and data. Storage, presentation, archiving, browsing, and searching are basic Montage functions applicable to most object types. Design analysis and module abstraction and description are conceptually higher-level functions also supported by Montage.

Montage design paradigms draw heavily from the field of Software Engineering, which faces very similar challenges. We are building the basic Montage framework on the Eclipse [23] platform, in order to leverage and build upon the many integrated development environment (IDE), resource sharing, and system design capabilities of Eclipse.

New levels of abstraction in experiment definition are a key component of Montage. In the original DETER testbed, experimenters had to specify a number of different types of resources in great detail. These included

- Computational elements such as physical or virtual hosts, and the complete "network plumbing" configuration of each.
- Elements of a network environment, including network topology, router and switch nodes and their configurations.
- Hidden facility nodes that perform traffic shaping to simulate real world network conditions, delays, throughput limits, etc.

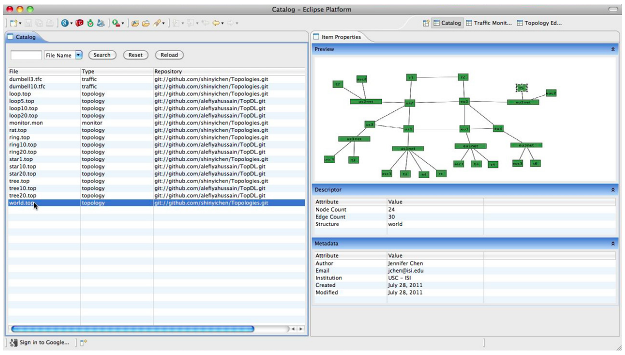
In addition, experimenters had to specify in detail the experiment elements running within the network, and, for each element, information such as host operating systems, guest operating systems for VMs, application software, and logging and other infrastructure software typical of real systems.

After completing this specification, experimenters had to deploy on their designed experiment a number of fixtures such as traffic generators, tools for running experimental procedures and collecting result data, and experimental configuration such as malware to be observed and cyber-defenses to be tested.

Further, each experimenter tended to do their own scenario construction largely from the ground up, with limited leverage of others' work in defining experimental apparatus. In contrast, Montage includes an extensive library function, capable of storing and cataloging both basic building blocks and composed structures contributed by prior experimenters. With Montage, experiments can be highly modular and explicitly structured for reuse as shown in **Fig. 1**.

Although the detail-oriented "expert mode" is still available, we expect most researchers to use Montage's facilities for defining an experiment more abstractly and at higher level. For example, an earlier experiment may already have defined an apparatus that simulates a handful of large enterprise networks connected over the public network, a number of ISP networks, and home computers.

Thus far, the description of Montage is analogous to an IDE with source code repositories, modules, libraries, facilities for combining them, with shared storage, versioning, and change control. Montage also provides further critical facilities analogous to an IDE:



File	Type	Repository
dumbell3.tfc	traffic	git://github.com/shinyichen/Topologies.git
dumbell10.tfc	traffic	git://github.com/shinyichen/Topologies.git
loop.top	topology	git://github.com/shinyichen/Topologies.git
loop5.top	topology	git://github.com/alefiyahussain/TopDL.git
loop10.top	topology	git://github.com/alefiyahussain/TopDL.git
loop20.top	topology	git://github.com/alefiyahussain/TopDL.git
monitor.mon	monitor	git://github.com/shinyichen/Topologies.git
rat.top	topology	git://github.com/shinyichen/Topologies.git
ring.top	topology	git://github.com/shinyichen/Topologies.git
ring10.top	topology	git://github.com/alefiyahussain/TopDL.git
ring20.top	topology	git://github.com/alefiyahussain/TopDL.git
star1.top	topology	git://github.com/shinyichen/Topologies.git
star10.top	topology	git://github.com/alefiyahussain/TopDL.git
star20.top	topology	git://github.com/alefiyahussain/TopDL.git
tree.top	topology	git://github.com/shinyichen/Topologies.git
tree10.top	topology	git://github.com/alefiyahussain/TopDL.git
tree20.top	topology	git://github.com/alefiyahussain/TopDL.git
world.top	topology	git://github.com/shinyichen/Topologies.git

Attribute	Value
Node Count	24
Edge Count	30
Structure	world

Attribute	Value
Author	Jennifer Chen
Email	jchen@isi.edu
Institution	USC - ISI
Created	July 28, 2011
Modified	July 28, 2011

Fig. 1 Screenshot of an experimenter using Montage to view a catalog of experiment components, and select and view a network topology displayed visually.

- Tools for interpreting experimental data to yield information that expresses experimental results in terms of the experiment's model and the abstractions that helped define the apparatus.
- Mechanisms for "realizing" an abstract, modular experiment definition by allocating and configuring real network and computing elements.

The next sections describe our work on model-based experimentation, and on advances in realizing and running experiments at scale. That work is directly reflected into the Montage methodologies and tools mentioned above.

4.2 Model Based Experimentation

As DeterLab magnifies the scale and realism of scenarios available to experimenters, the challenges of defining appropriate experiments and learning from their execution dramatically increases. One long-term approach to this problem is to re-conceive the methodology of how cyber-security experiments are defined.

The basis for this approach lies in adopting basic ideas from other experimental sciences that are more mature than experimental cyber-security is at present. The conceptual starting point of an experiment is a real-world situation that displays an interesting problem that is inconvenient to investigate *in situ*. Instead, we define a conceptual model of the situation, and begin to define laboratory activity that allows us to construct in the lab a physical (or chemical, or biological, or informatic) model of the real-world situation. The model, in turn, serves as a specification for an experimental scenario that the experimenter will observe or interact with. As observations and interactions proceed, inferences from lab observations to the real world suggest where analogous modifications may create analogous results.

It is crucial to understand that this methodology is effective for some, but not all, experimental purposes. The methodology is appropriate *only when it is possible to construct a model* that is accurate in the necessary ways. If the experimenter lacks sufficient knowledge about the actual scenario to model it accurately, the methodology will generally lead to incorrect or misleading results. Consequently, a key question is whether the problem being studied is well enough understood that an accurate model exists or can be constructed. If this is *not* true, then the researcher must take a different approach[*1].

This model-based approach requires new cyber-security methodology and new experiment support tools. These tools are integrated into the already-described experiment lifecycle facilities, but are very different in nature from those previously described.

Rather than being focused on low-level experiment definition, these modeling tools are oriented to defining semantics for an experiment and its results, validating an experimental apparatus, and extracting understanding from results. Such tools might include:

- Semantic mechanisms to capture the intent of the experi-

[*1] Although an aspect of our research is concerned with determining when this methodology is and is not appropriate, we do not discuss this here due to space limitations.

menter;

- Support for monitoring this the semantic correctness of an experiment as it executes;
- Abstraction and modeling techniques for experiment design, realization, visualization, and analysis.

As in any scientific or engineering discipline, the greatest challenge often lies in a) creating an appropriate representation of the object for study, representative across the measurement dimensions that matter, and b) knowing whether or not you have succeeded in doing so.

While the most general case of this problem is very hard, we are working within DETER to develop restricted, purpose-specific methodologies targeting specific classes of cybersecurity experiment. We approach this through a set of Model Based Scenario development techniques, in which existing prototype models are tuned and specialized capture the behavior of different dimensions of cyber security experiments.

Using the workbench and tools that we are investigating, an experimenter is able to refine these existing, customizable, models into more concrete templates or recipes, which can be used to guide experiment scenario definition and data analysis. The specialization is base on a *knowledge discovery* procedure (shown in the middle of **Fig. 2**) that is derived from the model and its various components, together with formalized assumptions such as behavioral invariants or functional constraints. In other words, we are working towards a shift in methodology where new tools assist experimenters in rigorous construction, execution, and interpretation of *semantically validated* experiment design and execution.

A simple example is the development of a model state space for execution of (and potential attack on) a communication protocol. A variety of data (packet dumps, web server logs, auth logs) can be normalized for input into analysis and visualization tools that assist the experimenter in mapping from actual events to expected behaviors. **Figure 3** shows a conceptual view of the model state space, with various possible paths through it; a path to the "success" node would be expected results of experiment execution (visible in detail in event logs), while other paths indicate a violation of an assumption about correct behavior, which may be detectable sign of an attack or malfunction (accompanied by a particular reason for the violation, attributable to event logs).

Model based experimentation takes on an increasing importance when designing experiments that span both cyber and physical elements. The physical components are likely based in some set of models (real world, empirical, or theoretical). In order to capture the interactions and relations between the cyber and physical, it will be necessary to compose models. Recent work in Secure Smart Grid Architectures [24] argues that:

> "An analysis of the cyber-physical security of a smart grid architecture must focus on the impact of faults and interactions that cross domains rather than the localized response that might be seen in traditional penetration testing. This requires a capability to model large scale response to cyber-attack, as well as to perform modeling or simulation of the physical components of a system."

We view current research efforts such as the Smart Grid and other emerging cyber physical domains as new use cases for examining and validating the evolving features and capabilities of the DeterLab that we are developing as part of the DETER project research program.

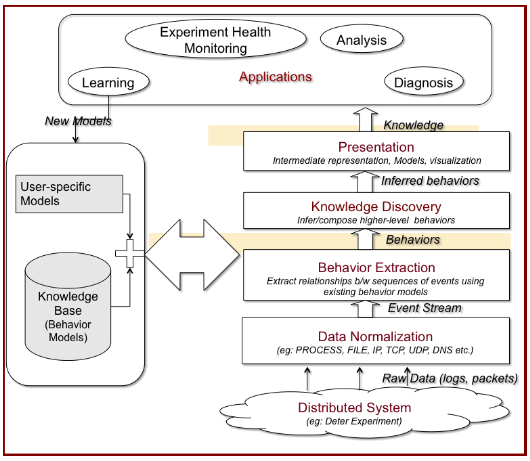

Experiment Health Monitoring
Analysis
Learning
Applications
Diagnosis
New Models
Knowledge
Presentation
Intermediate representation, Models, visualization
Inferred behaviors
User-specific Models
Knowledge Discovery
Infer/compose higher-level behaviors
Behaviors
Behavior Extraction
Extract relationships b/w sequences of events using existing behavior models
Knowledge Base (Behavior Models)
Event Stream
Data Normalization
(eg: PROCESS, FILE, IP, TCP, UDP, DNS etc.)
Raw Data (logs, packets)
Distributed System
(eg: Deter Experiment)

Fig. 2 Development of experimental knowledge from experiment data.

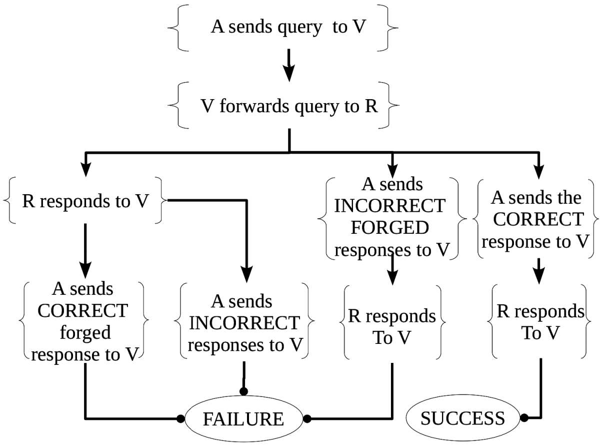

A sends query to V
V forwards query to R
R responds to V
A sends INCORRECT FORGED responses to V
A sends the CORRECT response to V
A sends CORRECT forged response to V
A sends INCORRECT responses to V
R responds To V
R responds To V
FAILURE
SUCCESS

Fig. 3 An example of a semantic model for an experiment.

4.3 Containers: Scale-up and Flexible Fidelity

Our continuing work on scalability is based on the observations (summarized in Section 3.4) about trade-offs between the fidelity or realism of a computational element in DeterLab, and the scale of network and computing resources required to realize a computational element. However, re-usability is also an important goal for the ease of use of DeterLab tools for constructing an experimental apparatus. By adding new types of computational element (conventional VMs, QEMU lightweight VMs, processes on conventional OSs, QEMU processes, individual threads of execution), each of which can be used to model a node in a simulated network, we added both flexibility and complexity to the methods of constructing an apparatus.

To manage complexity and increase ease of construction, we are developing an apparatus framework centered on an abstraction that we call a "container" [21]. In this, a container is the fundamental building block for realizing elements within an experiment scenario. Containers are recursive. A single container may support one or multiple components (elements) within a scenario, and implements an abstraction layer that hides the details of

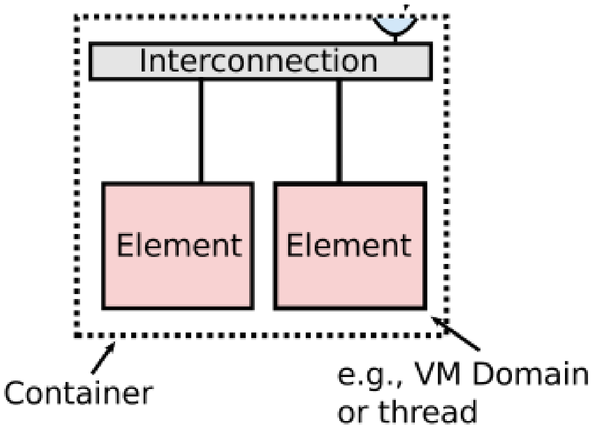


Fig. 4 A simple container of two basic computing resources.

the inner components when that container is itself placed inside another container. **Figure 4** shows a simple container that contains 2 concrete computing elements, such as a VM or thread, and no other containers. Abstraction is provided by the container's communication mechanism, which both connects the contained elements with one another, and also presents an entry/exit point for communication into the container; the communication mechanism advertises to other containers the properties of its container.

Containers are an implemented capability within the DETER facility today. The container system has been used successfully to support large multi-resolution experiments. One example modeled a worm/botnet/DDOS scenario of over 50,000 nodes. In the scenario, some nodes (attacked servers, DDOS defense mechanisms) were modeled with extremely high fidelity, while others (compromised end-node hosts) were modeled with only the necessary fidelity to implement the bot-driven DDOS attack. This demonstrates the value of the scalable multi-resolution approach to experiment construction enabled by containers.

Although the basic containers system is deployed, further work remains. As one example, creation of the above scenario involved manually matching each element in the desired network topology with a specific container. We are working to automate this process by developing container selection and specification algorithms driven directly by behavioral models of experiment elements, so that appropriate element fidelity, and thus appropriate container choices, can be determined automatically.

4.4 Additional Directions

The three previous sections have outlined some key areas of our current research work. Our research program includes additional active topics as well as planned future work.

Risky experiment management is one area of prior work that also occupies a place in the roadmap. To put into practice the management approach described in Section 3.2, we will need to (a) develop DeterLab facilities for an experimenter to develop and refine specifications of their experiment's requirements for Controlled Internet Access, and (b) develop automation tools to create an experiment-specific gateway node. The automation tools will need to both implement the experimenter's requirements, and also implement DeterLab's constraints defined in the T1/T2 approached described in Section 3.2.

This future elaboration of risky experiment management depends in part on the results of two areas of current research activity. Our modeling and specification work (described in Section 4.2) will provide key elements of the experimenter facility to define constraints and invariants on the experiment's communication via controlled internet access. Containers (described in Section 4.3) will enable DETER project research staff to create reusable building blocks for gateway implementation, each with advertisements that will assist the automation tools in constructing a container to serve as a gateway node that implements the required controls for controlled internet access as needed by the particular experiment.

A second part of the research roadmap is the support of *multi-party experiments*, a new form of DeterLab experimentation. A multi-party experiment is one in which the experimental apparatus is built from sub-components that are logically isolated, yet interconnected to create the whole, as is the real Internet. In a multi-party experiment each participant has complete information only about their own portion of the system, with only partial information about other sub-components. This form of experiment can be used to model several different kinds of cyber-defense situations: adversarial situations (e.g., red-team/blue-team exercises); realistic forensic or defense scenarios (e.g., attack target with limited information about attacker); or partial collaboration situations in which separate organizations collaborate on defense without granting full visibility to collaborators.

DETER's current implementation of multi-party experiments is based on DETER federation [9]. Support for multi-party experimentation will depend on the current full-production federation capability in DeterLab, and the results of several areas of current DETER research: modeling and specification work (described in Section 4.2) to state constraints and invariants on activities of each party; and containers (described in Section 4.3), which are essential to scale out each party's sub-apparatus to realistic proportions needed for the types of multi-party experiments currently envisioned.

4.5 Integrating the Pieces: Towards a New Experimental Cybersecurity Research Paradigm

The above current and future research roadmap provides the foundation for our program goal of new science based experimental cybersecurity. Our focus is extending DeterLab with new capabilities resulting from work in these areas, as well as integrating the new and existing capabilities. The integration is critical, including functional integration with the new Montage workbench; but more important is integration into a new *methodology* for the experiment lifecycle. Five of several possible lifecycle phases are illustrated in **Fig. 5**:

These are

- A new methodology for specifying experiments, including model-based specification, and elements of previous experiment descriptions;
- New tools to completely flesh out the structure of an experiment, with only the essential elements and abstractions;
- New technology for realizing the conceptual structure of an experiment, by embedding it in a subset of DeterLab's real

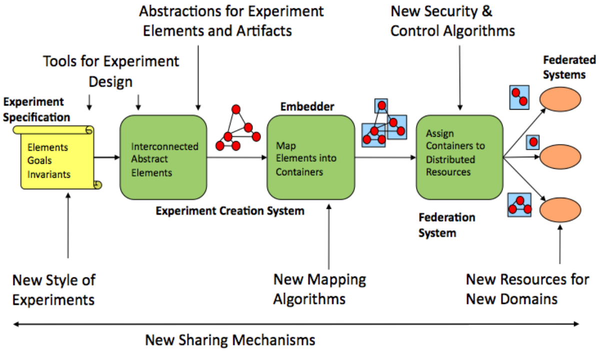

Abstractions for Experiment Elements and Artifacts
New Security & Control Algorithms
Tools for Experiment Design
Federated Systems
Experiment Specification
Elements
Goals
Invariants
Interconnected Abstract Elements
Embedder
Map Elements into Containers
Assign Containers to Distributed Resources
Experiment Creation System
Federation System
New Style of Experiments
New Mapping Algorithms
New Resources for New Domains
New Sharing Mechanisms

Fig. 5 New cyber-security research methodologies.

and virtual resources for computation and networking;

- New facilities and new controls that enable larger scale and more flexible use of federated systems and domain-specific resources – especially domain-specific resources that are available via federation; and
- Across all of these areas, new mechanisms and facilities to share experiment building blocks among experimenters, who can accelerate their experiment-creation work using the results and knowledge gained by previous work in DeterLab.

As we gain experience with this integration, we expect that both the DETER research team and the larger community of DeterLab experimenters will develop new experimental methodologies that can help to accelerate the pace of cyber-security innovation, and also dramatically improve the scientifically demonstrated effectiveness of innovations as they move from the lab into practical use.

Acknowledgments This paper builds on efforts at USC/ISI over the past 8 years and would not have been possible without the contributions of the entire DETER team. We particularly acknowledge Bob Braden, Ted Faber, Alefiya Hussain, Jelena Mirkovic, and Mike Ryan for their strong and multiple contributions.

References

[1] Benzel, T., Braden, B., Faber, T., Mirkovic, J., Schwab, S., Sollins, K. and Wroclawski, J.: Current Developments in DETER Cybersecurity Testbed Technology, *Proc. Cybersecurity Applications & Technology Conference For Homeland Security* (*CATCH 2009*) (Mar. 2009).

[2] Benzel, T., Mirkovic, J., et al.: The DETER Project – Advancing the Science of Cyber Security Experimentation and Test, *IEEE HST 2010 Conf.*, Boston, MA (Nov. 2010).

[3] Heelan, S.: Vulnerability Detection Systems: Think Cyborg, Not Robot, *IEEE Security and Privacy*, special issue "The Science of Security," Vol.9, No.3 (May/June 2011).

[4] Hardaker, W., Kindred, D., Ostrenga, R., Sterne, D. and Thomas, R.: Justification and Requirements for a National DDoS Defense Technology Evaluation Facility, Network Associates Laboratories Report (July 2002).

[5] Bajcsy, R., Benzel, T., Bishop, M., Braden, B., Brodley, C., Fahmy, S., Floyd, S., Hardaker, W., Joseph, A., Kesidis, G., Levitt, K., Lindell, B., Liu, P., Miller, D., Mundy, R., Neuman, C., Ostrenga, R., Paxson, V., Porras, P., Rosenberg, C., Tygar, J.D., Sastry, S., Sterne, D. and Wu, S.F.: Cyber defense technology networking and evaluation, *Comm. ACM*, Special issue on "Emerging Technologies for Homeland Security," Vol.47, No.3, pp.58–61 (Mar. 2004).

[6] Weaver, N., Hamadeh, I., Kesidis, G. and Paxson, V.: Preliminary results using scale-down to explore worm dynamics, *Proc. 2004 ACM Workshop on Rapid Malcode*, pp.65–72 (2004).

[7] Porras, P., Biesemeister, L., Levitt, K., Rowe, J., Skinner, K. and Ting, A.: A hybrid quarantine defense, *Proc. ACM WORM*, Washington, DC (Oct. 2004).

[8] Teoh, S.T., Zhang, K., Tseng, S.-M., Ma, K.-L. and Wu, S.F.: Combining visual and automated data mining for near-real-time anomaly detection and analysis in BGP, *Proc. ACM VizSEC/CMSEC-04*, Washington, DC (Oct. 2004).

[9] Faber, T. and Wroclawski, J.: A Federated Experiment Environment for Emulab-based Testbeds, *Proc. Tridentcom* (2009).

[10] Schwab, S., Wilson, B., Ko, C. and Hussain, A.: SEER: A Security Experimentation EnviRonment for DETER, *Proc. DETER Community Workshop on Cyber Security Experimentation and Test* (Aug. 2007).

[11] Emulab Testbed Web site, available from ⟨http://www.emulab.net⟩.

[12] DeterLab Testbed wiki, available from ⟨https://trac.deterlab.net/wiki/Topologies⟩.

[13] Hussain, A., Schwab, S., Thomas, R., Fahmy, S. and Mirkovic, J.: DDoS Experiment Methodology, *Proc. DETER Community Workshop on Cyber Security Experimentation and Test* (June 2006).

[14] White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M., Hibler, M., Barb, C. and Joglekar, A.: An Integrated Experimental Environment for Distributed Systems and Networks, *Proc. OSDI 2002* (Dec. 2002).

[15] Bavier, A., Bowman, M., Chun, B., Culler, D., Karlin, S., Muir, S., Peterson, L., Roscoe, T., Spalink, T. and Wawrzoniak, M.: Operating System Support for Planetary-Scale Network Services, *NSDI '04* (May 2002).

[16] Peterson, L. and Wroclawski, J. (Eds.): Overview of the GENI Architecture, GENI Design Document 06-11, Facility Architecture Working Group (Sep. 2006).

[17] Barford, P. and Landweber, L.: Bench-style Network Research in an Internet Instance Laboratory, *Proc. SPIE ITCom*, Boston, MA (Aug. 2002).

[18] Ostrenga, R., Schwab, S. and Braden, R.: A Plan for Malware Containment in the DETER Testbed, *Proc. DETER Community Workshop on Cyber Security Experimentation and Test* (Aug. 2007).

[19] Song, D., Brumley, D., Yin, H., Caballero, J., Jager, I., Kang, M.G., Liang, Z., Newsome, J., Poosankam, P. and Saxena, P.: BitBlaze: A New Approach to Computer Security via Binary Analysis, *Proc. 4th International Conference on Information Systems Security, Keynote Invited Paper* (Dec. 2008).

[20] Wroclawski, J., Mirkovic, J., Faber, T. and Schwab, S.: A Two-Constraint Approach to Risky Cybersecurity Experiment Management, Invited paper at the Sarnoff Symposium (Apr. 2008).

[21] Faber, T., Ryan, M. and Wroclawski, J.: Building Apparatus for Multi-resolution Networking Experiments Using Containers, in submission.

[22] Vahdat, A., Yocum, K., Walsh, K., Mahadevan, P., Kostic, D., Chase, J. and Becker, D.: Scalability and Accuracy in a Large-Scale Network Emulator, *Proc. 5th Symposium on Operating Systems Design and Implementation* (*OSDI*) (Dec. 2002).

[23] Silva, V.: *Practical Eclipse Rich Client Platform Projects* (1st ed.), ISBN 1430218274 (Mar. 2009).

[24] Neuman, C. and Tan, K.: Mediating Cyber and Physical Threat Propagation in Secure Smart Grid Architectures, *Proc. 2nd International Conference on Smart Grid Communications* (*IEEE SmartGridComm*), Brussels (Oct. 2011).

[25] Maxion, R.A., Longstaff, T.A. and McHugh, J.: Why Is There No Science In Cyber Science [A Panel Discussion at NSPW 2010], *New Security Paradigms Workshop*, Concord, MA, USA (Sep. 2010).

[26] Benzel, T.: The Science of Cyber Security Experimentation: The DETER Project (Invited Paper), *Proc. Annual Cyber Security Applications Conference* (*ACSAC*), Orlando, Florida, USA (Dec. 2011).

Appendix

A.1 The DeterLab Facility

The DeterLab facility provides a general purpose, flexible platform for modeling, emulation, and controlled study of large, complex networked systems.

The DETER Project and the DeterLab facility are lead by the University of Southern California's Information Sciences Institute (USC/ISI), with support from the University of California Berkeley and Sparta Inc. The physical facility includes three computing clusters located at USC/ISI (Los Angeles), UC Berkeley, and USC/ISI's US east coast site (Arlington, VA). The DeterLab facility has been operational since early 2004. The core operating system of DeterLab has its origins in the Emulab software from the University of Utah [11]. This software base has

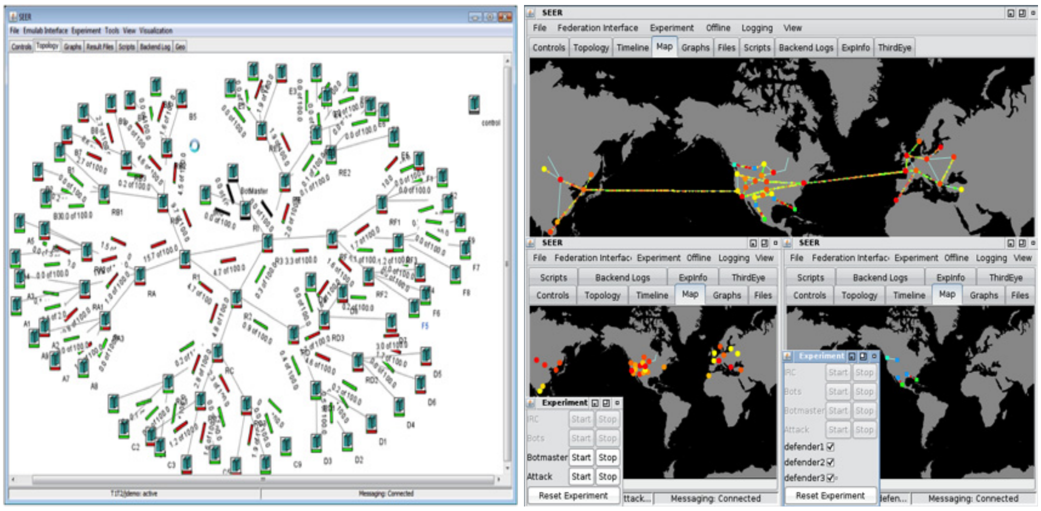

SEER
File Emulab Interface Experiment Tools View Visualization
Controls Topology Graphs Result Files Scripts Backend Log Geo
BotMaster
SEER
File Federation Interface Experiment Offline Logging View
Controls Topology Timeline Map Graphs Files Scripts Backend Logs ExpInfo ThirdEye
Experiment
IRC Start Stop
Bots Start Stop
Botmaster Start Stop
Attack Start Stop
defender1
defender2
defender3
Reset Experiment
Messaging: Connected

Fig. A·1 Control, Analysis and Visualization Tools – Topology Representation and Real-Time Traffic Visualization with Menu Options.

been extended to include low-level capabilities essential to security experimentation, together with the addition of the many methodological frameworks and tools described here.

The facility is both time-shared and space-shared, and is capable of supporting multiple simultaneous experiments at any given time. DeterLab offers a number of specialized environments for control, analysis, and visualization of the experimental process as well as a number of specific interfaces and tools, including GUIs, traffic generators, simulators, and traffic analyzers. A few of these environments and their interfaces are shown in **Fig. A·1**. Access to the DeterLab facility is open to academic, industrial, and government researchers worldwide and is granted through a lightweight approval process. Approved users are able to access the facility through a web interface.

A.2 The LACE Project

The DETER Project is engaged in a multi-site collaborative effort in networking and cyber security research and infrastructure with several Japanese institutions. Initial collaborators included the University of Southern California Information Sciences Institute in the United States and JAIST (Japan Advanced Institute of Science and Technology), NAIST (Nara Institute of Science and Technology), and the University of Tokyo in Japan, with sponsorship from the US National Science Foundation and Japan's NICT. As the project has developed, a more direct link to NICT has developed as well.

The collaboration spans three elements, each intended to include contributions from both the US and Japan. These are:

- Design and implementation of a federated system of ISI's DETER and JAIST's StarBED networking and cybersecurity testbeds. Building on shared interests in heterogeneity, federation, experimental science, and advanced research methodologies, the federation aims to provide a world-class facility that brings leading-edge tools and experimental methodologies to researchers in both countries.
- As drivers for the federation, two catalyst research projects as the first two concrete applications of the federated facilities.
- Exchanges and interactions to identify and support more extensive collaborations that leverage the shared infrastructure, our existing relationships, and complementary research interests.

To date activities are primarily in the area of the first element: creation of an integrated, federated research infrastructure spanning the two facilities.

Terry Benzel is Deputy Director for the Computer Networks Division at the Information Sciences Institute (ISI) of the University of Southern California (USC). She participates in business development, technology transfer and special projects with industrial and academic partners. She is the technical project lead for the Cyber Defense Technology Experimental Research (DETER) testbed projects funded by DHS, NSF and DARPA. The projects are developing an experimental infrastructure network and scientifically rigorous testing frameworks and methodologies to support the development and demonstration of next-generation information security technologies for cyber defense. Ms. Benzel has a joint appointment at the Marshall School of Business where she is a researcher at the Institute for Critical Information Infrastructure Protection. She is responsible for helping to develop Systemic Security Management as an open source body of work and developing public/private partnerships in information security research.

John Wroclawski is Director of the Computer Networks division at the University of Southern California's Information Sciences Institute, with responsibility for the strategic direction of this 40-member research organization. ISI's Computer Networks Division maintains active programs of research in areas such as Internet protocols and architecture, sensing and sensor nets, network and distributed system security, cyberphysical systems and the Smart Grid, and space systems networking. Wroclawski's personal research interests include the architecture, technology and protocols of large, decentralized communication systems such as the Internet, architectural aspects of cyberphysical systems, and the core principles of self-organizing and self-structuring architectures. At USC/ISI, Wroclawski also serves as chief scientist of the DETER Cybersecurity testbed, a major US DHS and NSF funded project aimed at transforming the effectiveness, rigor and scientific merit of experimental cybersecurity study through research into new methodologies for the field, together with deployment of the developed methodologies within a publically available advanced experimental infrastructure. In the broader community, he served from its inception on the planning group for US National Science Foundation's GENI project, co-chaired from 2006–2008 the working group charged with overall technical architecture of the proposed GENI facility, and continues as an active participant in the program.