The Science of Cyber Security Experimentation: The DETER Project

Terry Benzel

USC Information Sciences Institute
4676 Admiralty Way #1001
Marina del Rey, CA 90292
+1-310-822-1511

tbenzel@isi.edu

ABSTRACT

Since 2004, the DETER Cyber-security Project has worked to create an evolving infrastructure – facilities, tools, and processes – to provide a national resource for experimentation in cyber security. Building on our insights into requirements for cyber science and on lessons learned through 8 years of operation, we have made several transformative advances towards creating the next generation of DeterLab. 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 other researchers in the DeterLab user community. This paper describes the advances resulting in a new experimentation *science* and a transformed facility for cyber-security research development and evaluation.

Categories and Subject Descriptors

D.4.6 Security and Protection D.4.8 Performance Measurements - Modeling and prediction, Monitors.

General Terms

Algorithms, Management, Measurement, Performance, Design, Economics, Reliability, Experimentation, Security, Verification.

Keywords

Cyber-security, testbed, experimental research

1. INTRODUCTION: THE DETER PROJECT

DETER is a research project that is advancing cyber security research practices, by extending the methods, technology, and infrastructure required for scientific development of cyber-defense technology [1], [2]. Our research results are put into practice by operating and evolving DeterLab, transforming it from a basic hardware-focused network security testbed within the early phase of the project, through the shared experiment and testing lab of the middle phase of DeterLab use, towards a new kind of facility for cyber-security experimental science.

Our vision for the next iteration of DeterLab is a facility that is used as a scientific instrument, where researchers create knowledge and understanding through observation, modeling, and measurement of computer and cyber security threats and defenses. We are actively engaged in research and development to improve the scope and usability of this instrument, throughout the life cycle of an experiment – definition, execution, and interpretation. An additional and fundamental goal is that the scientific facility will be to support diverse research communities, and enable users to contribute to and leverage a common body of tools and knowledge. That leverage will enable researchers to build on one another's experimental work, with reproducible experiments and repeatable experimental procedures.

We believe that this evolution of DeterLab will enable shifting the science of cyber security experimentation towards rigorous design, construction, execution, and interpretation of experiments. Such a shift is required to advance the scale, pace, and power of cyber-security research, and to expand the research community and accelerate their work of developing the innovative, scientifically tested, and demonstrably effective new cyber-defenses required to meet challenges ranging from personal digital security to national-scale critical infrastructure protection.

In pursuit of this goal, the DETER project's research program has been influenced by several lessons learned from our experiences in evolving DeterLab, and from supporting and interacting with many researchers using DeterLab. Those lessons have driven several new areas of work on developing cyber-security science methods and technology, that is, the tools, techniques, methodology, resources, and facilities that are needed to provide DeterLab researchers with major advances in their scientific and experimental capabilities: repeatable, flexible, and variable experimentation, discovery, observation, and testing.

This paper provides: some background motivation and history for the DETER project; a review of our key advances in creating a science of experimentation, and lessons learned from these advances; a description of the current DETER research program and roadmap; and a prognosis for the use in DeterLab of the resulting innovations in cyber-security research practices and infrastructure, as we progress toward our vision of a new environment and community for science based cyber security experimentation and test.

2. BACKGROUND: MOTIVATION AND HISTORY

The DETER project's creation grew out of a set of related observations made within the computer and network security

ACSAC'11, Dec. 5-9, 2011, Orlando, Florida, USA.

research community, funding organizations, and security product companies:

- 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.
- Successfully and 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) also need large scale and highly varied testing.
- 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 test facility. 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 testing facilities with significantly greater resources and flexibility than the limited test environments of most innovators, and greater fidelity to real deployment environments. One 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 funders at NSF and 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, with more complex test fixtures, assembled 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 two NSF programs – Experimental Infrastructure Network (EIN) and Network Research Testbeds (NRT) – and by DHS. At the same time EIN and NRT co-funded the EMIST project, composed of researchers from Penn State, McAfee Research, ICSI, Purdue, Sparta, Inc., SRI International, and UC Davis. EMIST researchers were to use the DETER testbed, help build knowledge about researcher's needs based on experience working in the testbed, and build experience with existing testing tools used in the testbed. Together these efforts led to the success of the first phase of DETER, with 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. Both DETER researchers and community collaborators worked on research topics in the technology for supporting and enabling cyber-security research work: experiment automation, benchmarking, scaling via hypervisor usage, malware containment, and our initial work on federation [9], now a central component of DeterLab technology.

In the second phase of DETER, 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 lab composed of the underlying testbed resources, technology for using and managing the resources as test fixtures, and a growing variety of tools and services for experiment support, including the full-support release of the federation capability, and the first-generation experimenters' workbench, SEER [10].

With the technological maturity achieved in this phase, and the experience gained from supporting over 1000 researcher team members, the stage was set for DETER project activities to focus increasingly on research and development in the areas of cyber-security experimentation methodology, infrastructure, tools, resource expansion, utilization innovations, and other new methods of using DeterLab for scientific experimentation.

The experience during this phase included several lessons learned that provided important guidance for the current DETER research program. The balance of this paper describes five significant lessons learned, and outlines the current research program developed from the combination of our vision of an advanced scientific instrument, and lessons learned from the developing community.

3. LESSONS LEARNED: CHALLENGES FOR SCIENCE BASED EXPERIMENTATION

Here we describe several lessons learned from early experience with DETER. These lessons derive from observing norms of researcher activity that emerged in the 2nd phase of DeterLab use. Those norms included some basic terms and a typical workflow. Researchers working in DeterLab are called "users" or "experimenters". A team of users working together is called a "project." The focus of activity of a project is a construct called an "experiment" – a term that applies whether the construct is used in the strict sense as part of activity to prove or disprove a hypothesis, or the construct is used for broader purposes such as observing malware behavior, measuring the effect of counter-measures, or other forms of testing or observation that could contribute to hypothesis formation or demonstration of effectiveness of a counter-measure.

The concept of an "experiment" is broad. In DeterLab's vernacular the term "experiment" is used at minimum, to describe

the experimental apparatus or environment that users have constructed from computing and network resources, hosts, software, test fixtures, measurement tools, and other fixtures or components of a system that experimenters operate. Beyond this, "experiment" is also frequently used to describe the experimental procedures and experimenter activity to interact with an apparatus in operation, and to review results of operation – that is, the entire "experimental protocol" for a particular research activity.

A large portion of our learning consists of observations and user feedback about the process of designing, building, and carrying out an experiment. Experimenters typically work in a high-level workflow that consists of: initial design and construction of an experimental apparatus; exploratory operation of the apparatus during iterative construction; initial full test runs of operating the apparatus; reviewing and analyzing results of test runs; iterative re-working of the apparatus, towards a fully satisfactory apparatus that repeatably operates as expected, and generates output such as log data, network traces, etc.; analysis of output data to create experimental results. We refer to this workflow as the "experiment lifecycle." Our goal is to change this lifecycle from a manual human intensive *ad hoc* set of processes to a highly automated set of processes tied semantically to an experimenter's model and exploration motivations.

3.1 Lesson Learned: Experiment Construction Considered Difficult

DETER's original tools for construction of experimental apparatus were inherited from Emulab [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, and using them to create emulated networks within which an experimenter's activity took place. For our original set of EMIST researchers, this toolset was useful, because they were experienced researchers who valued the "expert mode" in which every detail of a test network could be specified. In addition, many of these early users were familiar in concept or in experience with other network testbeds.

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.

One reason for the complexity of network definition lay in the typical first step, preparing a file containing expressions in a specification language to define each individual network node. From "toy network" examples [12] one can extrapolate the level of detail required to specify non-trivial network topologies. In addition to the topology, each node may need to be assigned certain properties so that, e.g., some nodes may serve as background traffic generators or traffic delay nodes. Following topology definition and attribute specification, there are further steps: nodes acting as hosts need to be loaded with a boot image of OS and application software, often followed by the addition of other software, such as experiment-specific software, or packages for monitoring and logging host or network activity. All nodes must be network-configured with network interface devices, IP address, DNS and default route settings, etc.

In addition to the level of effort required, we also had a concern about methodology – in most cases, each experimenter had to do almost all the above activities, with very little leverage of previous experimenters' work, other than perhaps using published network definition files as a model or template. Our own work [13] to build a reference case of a DDOS experiment, designed for re-use and extension by others, was instructive and useful, but also served to highlight the large amount of required detail that was not central to some researchers' work, but was nevertheless pre-requisite to a functioning experiment.

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), we added the objective that a new experiment should be able to "stand on the shoulders of previous experiments, rather than standing on their feet".

3.2 Lesson Learned: Diverse and Flexible Federation Considered Valuable

DeterLab's federation capability was originally conceived out of our goal to expand the possible scale of an experiment by enabling an experiment to use not only DeterLab's physical computing and network resources, but also those of other testbed facilities, such as such as Emulab [14], PlanetLab [15], and GENI. [16] The DETER team set up federation arrangements and functionality with such testbeds, both as part of our own research on federation, and also to benefit experimenters seeking larger scale and/or more fidelity by introducing wide-area network topology into an experiment.

However, once the first-generation federation capability was used by DeterLab researchers, we learned of additional benefits sought by them, beyond DETER-managed federation with other testbed facilities. One type of additional benefit was the inclusion in an experiment of unusual or unique resources available in specialized facilities, for example: SCADA systems and embedded controllers, in national labs; supercomputing facilities in high-performance computing facilities; and rare, new, or very large scale networking gear available in labs set up for testing them, such as the University of Wisconsin WAIL [17] facility.

In addition to this "specialized" integration of computing and network resources outside DeterLab, some researchers also sought federate with their own facilities and/or those of their collaborators. Some additional degree of scale-up could be achieved by joining those more ordinary resources "down the hall" from the researcher with the larger scale resources in DeterLab.

These types of desired federation were beyond the scope of the original federation model of linkage between an external network testbed and the testbed underlying DeterLab. To support these types of federation – and others not yet conceived – we began to address issues of resource usage policy and access control, and enriched the federation mechanisms to support them, beyond the originally conceived testbed-to-testbed federation.

3.3 Lesson Learned: Experiment Isolation Considered Limiting

During this same period, we noted that changes in the threat landscape required stark changes to the DETER facility's initial conception of experimental environment. For example, the initial 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, which was fully isolated from the public network by a number of extremely rigid segregation measures [18].

This approach quickly proved limiting for cases where the software-in-the-wild has non-deterministic behavior; because in this case 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 some software of interest has properties that depend on test fixtures or test procedures – specific software, or specific behavior, or human factors – that are difficult to replicate at high fidelity. 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 both short-term expedient approaches to controlled internet access and longer-term approaches to flexibly manage this sort of interactions.

The short-term approach involves the use of an *ad hoc*, experiment-specific tunnel node as one node in an experiment apparatus, to permit other nodes to interact with outside systems. The tunnel node is specially prepared by DeterLab network operations, to implement network controls on outside access, but permit the interaction that the experimenter desired and that DeterLab management would permit. Naturally this approach is hardly scalable, requiring the use of scarce operations staff time, and relying on informally stated requirements for outside interaction.

Recognizing the limits of this approach, we also began work on a methodology for more flexibly managing CIA, by transforming a risky experiment into a safe experiment. Our work builds on a single, simple fundamental observation:

- If the behavior of an experiment is completely un constrained, 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.

3.4 Lesson Learned: A Requirement for Scale and Fidelity Flexibility

In the initial DETER testbed, the basic approach to scalability rested on scaling up the amount of available hardware, and on a fixed set of approaches to fidelity using individual nodes. By "fidelity" we mean that in a testbed there is a set of nodes that exist to model the behavior of nodes in the real world, and the testbed nodes' behavior in the simulated environment is behavior that is similar to real nodes in real environments. The fixed set of approaches to fidelity, in this classic approach, is a range from a single testbed node acting like a single real node, with high fidelity; to a single testbed node standing for a large number of internal network end-nodes.

In this approach, the maximum size of an experiment is essentially bounded by the number of hardware nodes. As we built out DeterLab, not only did we want to increase the scaling of the hardware, but we also recognized that many of our users' experiments did not require high fidelity in every part of the

experimental apparatus. 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 a number of 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 unified framework for the construction of composable experiment apparatus, using both some common building blocks and methods of composition with abstraction and re-use. Our approach to such a framework is to base on it on an abstract fundamental building block called a "container" which represents experimental elements at the same level of abstraction, and is the basic unit of composition for constructing an experimental apparatus. 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 with high resolution and fidelity for the things that matter most to them, and abstract out the less important elements;
- reduce the experimenter's workload of experiment apparatus construction, enabling larger scale apparatus with lower levels of effort.

3.5 Lesson Learned: From Experimental Apparatus to Experimental Data to Experimental Results

The previous four lessons learned were largely related to the static aspects of setting up an experimental apparatus: basic construction of an apparatus; use of fixtures for federation; use of fixtures to enable limited communication outside the testbed; and use of fixtures that support orders-of-magnitude experiment scale-up over that obtainable with more simplistic use of physical resources.

Other lessons learned were about the dynamic aspect of running an experiment. Early in the 2nd phase, we recognized the need for a workbench that experimenters could use to operate an experimental apparatus, feeding it input data and events, observing its operation, and adjusting the fixtures for collecting experimental data. The first-phase workbench, SEER [10], met that need to some extent. However, adoption of SEER also brought into focus a growing need for DeterLab experimenters: an approach to the "big data" problem. 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, a much larger set of output data to be analyzed from each experiment run.

Further, not only the size of data grew, 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 terms of the intended semantics of the experiment run, and ultimately to proceed from data analysis to actual experimental results: 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 mining experimental data to discover experiment results. 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 PROGRAM

Our current research program includes, but is not limited to, activities related to the above lessons learned. Current research is in some cases an outgrowth of work performed as part of our agenda to enrich the testbed with progressive enhancements that resulted in what we now call DeterLab. During the 2nd phase in which we were learning from DeterLab users, our own enhancement efforts included:

- First generation of federation capabilities [9];
- Risky experiment management and abilities to include outside communication [16];
- The first-generation "experimenter workbench" for managing an experiment in process, viewing its activity and results data [10].

In some cases, there was real synchronicity between our objectives and the needs of DeterLab experimenters. As described above, the first generation of federation capability arose from our desire to reach greater scale by using resources in other testbeds that we could link to; in addition, we learned that experimenters wished to link into their experiments some outside resources of their own, and/or specialized resources that they had access to. As a result, our research agenda (for federation with access control) was enriched with new use cases and additional requirements.

4.1 Experiment Lifecycle Management

Experiment lifecycle management is an outgrowth of work on our first generation workbench, SEER. Indeed, many of SEER's capabilities, including experiment monitoring and visualization, are carried over into the next generation workbench, the

Experiment Lifecycle Manager (ELM), albeit in a new usage paradigm.

One critical aspect of ELM focuses on the general concept 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, archival, browsing, and searching are basic ELM functions for managing an experiment's components – and allowing other researchers to access them – far beyond the original testbed approach of shell login and filesystem access. We are building this basic management framework on the Eclipse [23] platform, in order to leverage and build upon the many integrated development environment (IDE) capabilities of Eclipse.

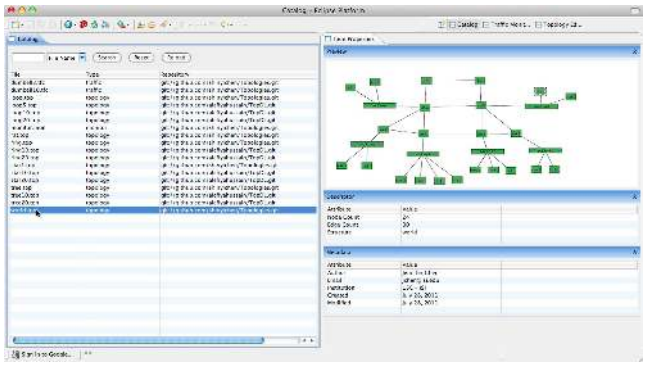

Figure 1: Screenshot of an experimenter using ELM to view a catalog of experiment components, and select and view a network topology displayed visually

New levels of abstraction in experiment definition are also a key component of ELM. In the original testbed approach, experimenters had to specify in detail a number of different types of resources:

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

In addition, experimenters had to specify in detail a number experiment elements running within the network and computational resources: host operating systems, guest operating systems for VMs, application software, and logging and other infrastructure software typical of real systems. Further, experimenters had to deploy on these systems a number of experimental fixtures such as traffic generators, tools for running experimental procedures and collecting result data, and often malware to be observed and cyber-defenses to be tested.

Perhaps most significantly, each experimenter tended to do their own apparatus construction largely from the ground up, with limited leverage of others' work in defining experimental apparatus. In ELM, all these types of experiment resources, elements, fixtures, and artifacts do of course need to be managed as individual objects, as building blocks for components of an experiment. More importantly, we're working on construction methods that include both the basic building blocks, and also structures of them that prior experimenters have contributed. In other words, with ELM, experiments can be highly modular, and explicitly structured for re-use as shown in Figure 1.

Although the detail-oriented "expert mode" is still available, we expect most researchers to use the newer facilities for defining an experiment's components abstractly, with requirements, constraints, and invariants, rather than specify directly and in every detail. 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. This apparatus, though conceived for use in worm spread, may nevertheless be described with meta-data that enables a later researcher to identify it as a suitable starting point for their work. The later researcher should be able to use the archived design, and state some new requirements and constraints relevant to their work, or specify some properties of specific experiment fixtures for input generation or monitoring. Without having to know other detail beyond their requirements, experimenters can describe an experiment apparatus entirely independent of its realization on computing and network resources.

Thus far, the description of ELM is analogous to an IDE with source code repositories, modules, libraries, facilities for combining them, with shared storage, versioning, and change control – all valuable advances from the early DETER testbed. However, ELM also provides other critical facilities analogous to an IDE:

- Mechanisms for "realizing" an abstract, modular experiment definition by allocating and configuring real network and computing elements.
- 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.

Following sections describe some of our work on advances in realizing and running experiments at scale, and on model-based experimentation that enables semantic analysis of results. That work is directly reflected into the ELM methodologies and tools mentioned above.

4.2 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 our new construction

methodology, a container is the fundamental building block. A single container may support one or multiple components (elements) of an experimental apparatus, and implements an abstraction layer that hides the details of the inner components, when that container is itself placed inside another container. Figure 2 shows a simple container that contains no other containers, containing only 2 concrete computing elements, such as a VM or thread. 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.

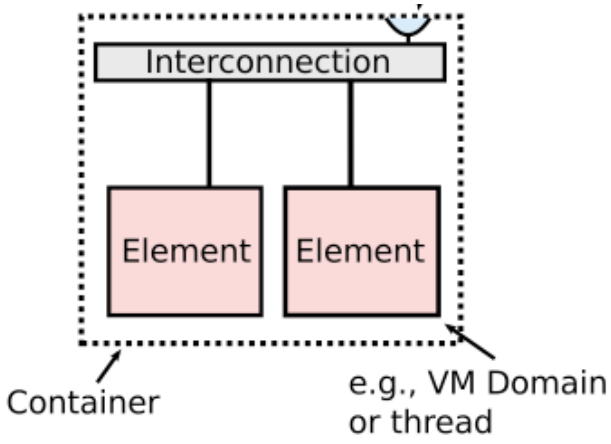

Figure 2: A simple container of two basic computing resources

Abstraction is a central point for continuing to expand the scalability options in DeterLab. Other researchers frequently create new techniques for scalable computing, or new methods of using virtualization or simulation, or performing a form of lightweight computation. Our goal going forward is to incorporate promising work in this area, defining a new abstract type of basic computing element, creating a standard interface for a containerized component based on each new technology, and expressing its tradeoffs explicitly, for use in construction tools.

Thus far, our containers work has been successful for scalability for multi-resolution experiments, and has illustrated the requirements for integrating container technology into the ELM workbench. In addition to the expansion of available basic computing elements described above, there are several areas of ongoing work.

Apparatus construction: We demonstrated feasible levels of scalability and complexity by creating several experiment apparatus, the largest containing over 600 components that were then realized on 8 physical computers and is capable of representing 50,000 computers at a coarse granularity in a specific scenario. The low end of this spectrum would be a modest sized mixed-resolution experiment. The high end would be a large experiment in which all but a few elements had low fidelity. However, construction involved manually matching each element in a desired network topology with a specific container. Clearly, there is promising work to do in automating this process, and integrating containers into the apparatus construction mechanism of our new workbench, ELM.

Re-usability and embedding: ELM provides the ability of experimenters to archive experimental apparatus definitions, or components of them, and for other experimenters to use these archived items as building block for the construction of a new apparatus definition. The definitions can then be used with core DeterLab embedder capabilities for realizing the definition in a real apparatus composed of DeterLab network and computing resources. Again, there is work to do building the workbench technology for containers being one of the objects that can be archived and reused. Likewise, there is work to do with the embedder, to automate realization with little input from experimenters, while also giving experimenters visibility on embedding so that they can vary some of resource utilization or vary the fidelity/scale tradeoff points for a specific apparatus.

4.3 Model Based Experimentation

As described in Section 3.5, DeterLab experimenters have a "big data" problem that will only grow as DeterLab magnifies the scale available to experimenters, and the breadth of tools for collecting experimental data. Our approach to this problem has been to completely re-conceive the methodology for how cyber-security experiments are defined in order to yield data to be analyzed for information that comprises experimental results.

The basis for this approach is no more or less than 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* in the real world. 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. Part of the function of this model is to serve as a design for an experimental apparatus that the experimenter will observe or modify in order to make inferences from lab observations to the real world where analogous modifications may create analogous results.

This common methodological framework is, however, somewhat unusual for some areas of computer science, where much research is in fact *in situ* – modify an actual computational or communication system to observe whether it better meets the researcher's goal for speed, efficiency, data access rates, power utilization, etc. Cyber-security research, however, is very definitely model-based. The real world has a host of large-scale systems with complex structures with vulnerabilities and potential mitigations. Where experimental modification of real systems (for example, induced cyber attack) is not feasible, we use a lab environment to create a model, an apparatus to approximate the model, experimental procedures to observe or induce changes in the apparatus, and so on.

However, the early stage use of the DETER testbed was not significantly model-based. Figure 3 is somewhat whimsical but accurate view of experimentation in which modeling is entirely mental, with no externally visible relation between the model and an *ad hoc* constructed apparatus, or its operation. The lab procedures are in fact quite valuable to the researcher, but *ad hoc*, and difficult to document or to be repeated by others. Nowhere is the *ad hoc* nature more evident in the research's unique ability to pore over network traces to find variations in worm behavior that may be attributed to worm propagation countermeasures.

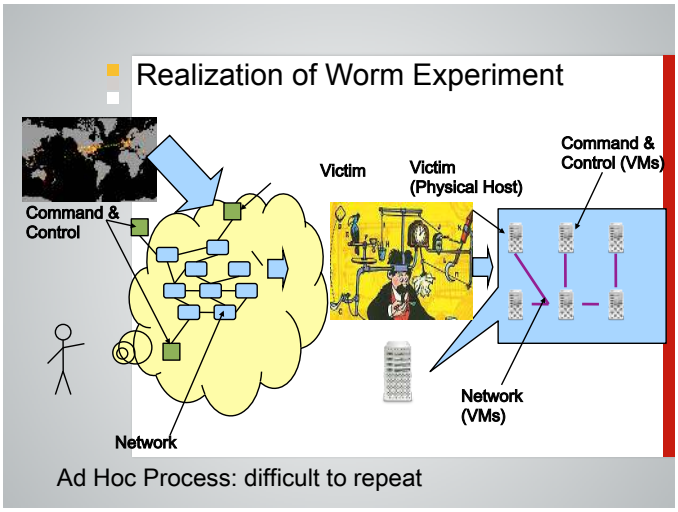

Figure 3: An informal experiment model leading to an *ad-hoc* experiment apparatus

The DETER research program includes work to assist researchers in defining several model-related structures that become part of the methodology for building experimental apparatus, defining experimental procedures, and analyzing experimental data for expected (or unexpected) patterns or changes predicted by the model or a hypothesis derived from it. One purpose of modeling and semantic definition techniques is to *define specific measurements and expectations* to be sought for in the results of an experiment's operation in the lab.

This model-based approach requires new cyber-security methodology and new lab technology, integrated into the already-described experiment lifecycle facilities, but oriented to defining semantics for an experiment and its results, validating an experimental apparatus, and extracting understanding from results. Several types of tools under investigation can potential benefit:

- semantic mechanisms to capture the intent of the experimenter;
- support for monitoring this intent and distributed execution with breakpoints;

abstraction and modeling techniques for experiment design, realization, visualization, and analysis.

The use of these tools is inherently iterative, shown in Figure 4. An experimenter **defines** a model, an apparatus to implement it, procedures to operate it; then runs the experiment by operating the apparatus, **executing** software or manual steps to perform experimental procedures; the resulting data is **interpreted** to extract information, which can then be used to iterate on the experimental apparatus, measurement scheme, or procedures.

To date, most experimentation presumed the existence of some form of model of the system under test that the experimenter uses to map his experiment objectives onto an apparatus in an experimental facility such as DeterLab. While this has often been true for the low-abstraction-level tasks of defining network topologies and traffic generator configurations, the lack of rigor in the initial steps often undercut the efficacy of the entire experimental process by providing little guidance or expectation for the resulting experimental data, and no ability for knowledge-based iteration.

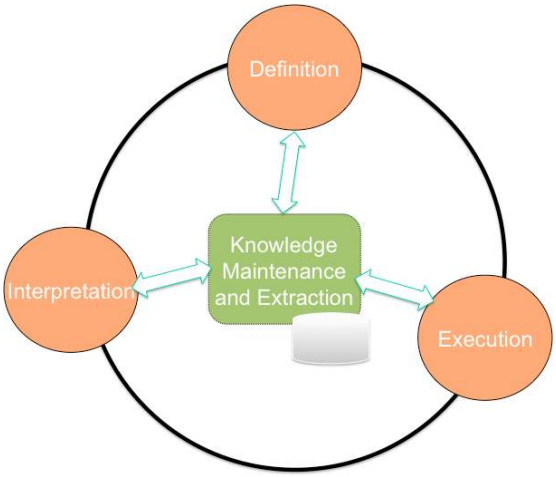

Figure 4: The iterative aspect of the experiment lifecycle

As in any scientific discipline, often the greatest challenge lies in creating an appropriate representation of the object for study, representative across the measurement dimensions that matter, while carefully documenting the simplifying assumptions and abstractions that are made in the process. While the most general case of this problem is very hard, we are working to extend DeterLab's experimenter support back into the early reasoning process of experiment design. We approach this through a set of Model Based Scenario development techniques, in which experiments are constructed from a basis in general models that capture the behavior of different dimensions of cyber security experiments.

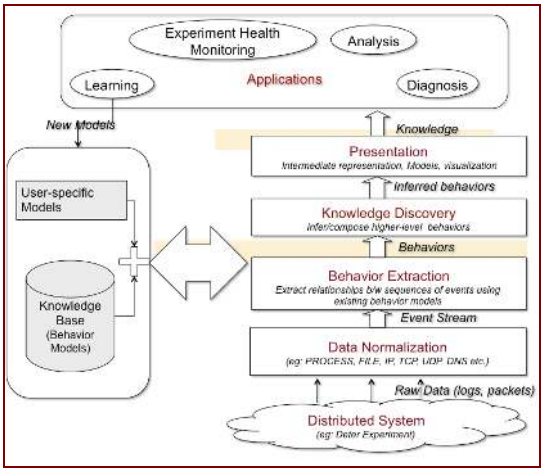

Figure 5: Development of experimental knowledge from experiment data

Using the workbench and tools that we are investigating, an experimenter may be able to refine the models into apparatus templates or experiment-procedure definition or recipes, which can be used to plan data analysis. The analysis would not be bottom up or *ad hoc* pattern based, but rather following a *knowledge discovery* procedure (shown in the middle of Figure 5) that is derived from the model and its various components and formalized assumptions such as expected 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.

During the specification phase of an experiment, invariants associated with the model will be used to verify that the experiment being developed is internally consistent. During execution, invariants will be used to ensure that the intended experimental semantics are realized (*validity management)*. Finally, invariants will be invoked as part of the interpretation and visualization of results – providing methods for the experimenter to tackle large amounts of data in order to determine whether an experiment run produced expected results or potentially interesting unexpected results.

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 6 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).

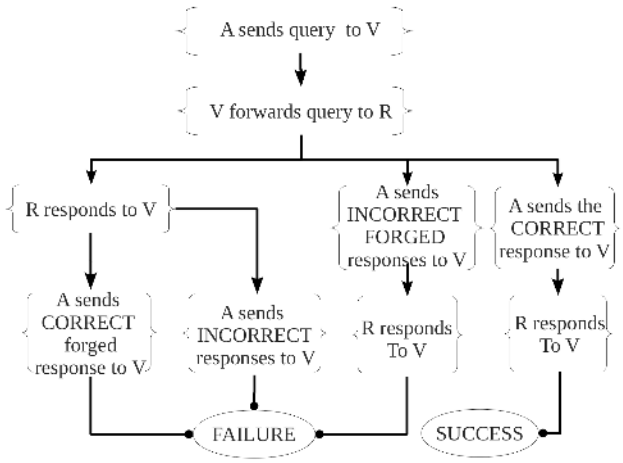

Figure 6: An example of a semantic model for an experiment

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."

The Smart Grid security team at USC-ISI and JPL are currently using DeterLab to develop a secure smart grid architecture. They ave created a taxonomy of cyber and physical threats and are exploring federation of DeterLab with other labs and testbeds that provide physical simulation and emulation tools for modeling the systemic response of the grid. Such experiments will span multiple sites and will enable the use of specialized resources to participate in large-scale experiments.

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.

4.4 Additional Directions

The three previous sections have outlined some of the key areas of our current research work, some of which was guided by lessons learned, in addition to the results of our own research. The research program also includes areas for future work, in which we will extend the results of earlier topics by applying to them some of the advances that we are making in our current research 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 3.3, 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 3.3.

For risky experiment management, this future elaboration will depend on the results of two areas of current research activity. The modeling and specification work (described in Section 4.3) will provide key elements of the experimenter facility to define constraints and invariants on the experiment's communication via controlled internet access. The container work (described in Section 4.2) 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 elaboration of prior work on federation, to support a new form of DeterLab experimentation. A multi-party experiment is one in which the experimental apparatus is built from sub-components that are federated to create the whole, and each federant has complete information only about their own sub-component, 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.

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 4.3) to state constraints and invariants on activities of each party; and the container work (described in Section 4.2), which is 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 areas of current research and future research roadmap provide the foundation for our program towards new science based experimental cybersecurity. Our focus is work is extending DeterLab 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 ELM workbench; but more important is integration into a new *methodology* for the experiment lifecycle. Five of several possible lifecycle phases are illustrated in Figure 7:

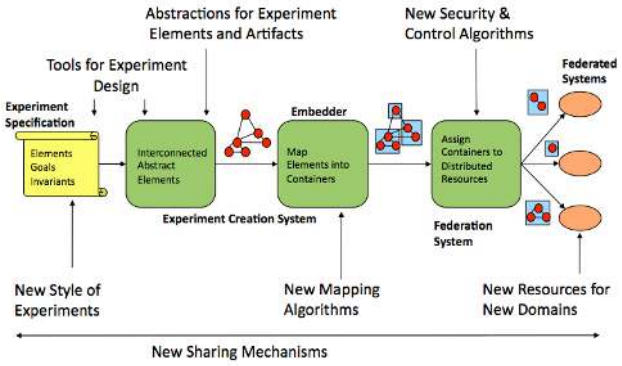

Figure 7: New cyber-security research methodologies

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 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 we and DeterLab experimenters will develop cyber-security experimentation 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.

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 the US National Science Foundation (contract number CNS-0831491). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of Homeland Security or the Space and Naval Warfare Systems Center, San Diego.

5. SUMMARY

The DETER project is continuing with its research program, much of which has been described in brief in this paper. Our research target continues to be the technology and methodology needed for the practice of cyber-security research to become an experimental science, with rigorous experiment design, construction, execution, interpretation, sharing, and repeatability of experiments. We believe that such a transformation of cyber-security research is needed to expand and accelerate the research community's efforts.

Expansion and acceleration are becoming increasingly important, as inter-connected cyber systems continue to expand into nearly every area of human infrastructure, from critical infrastructure for communication, finance, transportation, and power, to networked computing systems that are intimately integrated into our lives: our vehicles, our handheld wireless personal computing and communications ("smart phones"), regulation of our homes' utility use ("smart meters"), remote control and regulation of our homes' fixtures ("smart grid") including safety-critical appliances, as well as our healthcare, and even our bodies, with networked controllers for implanted medical devices.

From national and international security to personal safety, the number and variety of targets continues to increase along with an accelerating expansion of the total attack surface available to adversaries who have an increasingly powerful portfolio of tools and customers. When the DETER project started, cyber-security technology development and technology transfer were often resource intensive, and often lacking in proactive approaches for asset protection to be sufficient to increase the level of cyber defense of critical assets. In the past 8 years, we have built some significant infrastructure for the development and test of new cyber-defenses. As we look ahead to the coming years, we expect that transformation of research tools and methods will contribute to the much needed expansion and acceleration of research, which can lead to an accelerated pace of deployment of scientifically tested and effective cyber-defenses.

Continuing maturation of the DeterLab facility is necessary, but so is the accelerated growth of a cyber-security research and test community that can rapidly leverage one another's work. In addition to the research and methods described in this paper, further development of the cyber-security experiment science community is a critical shared responsibility of the larger cyber-security community.

6. ACKNOWLEDGEMENTS

This paper builds on efforts at ISI over the past 8 years and would not have been possible without the contributions of the entire DETER team. Special recognition is given to: John Wroclawski for setting an ambitious research program; Ted Faber, Mike Ryan, Alefiya Hussain and Jelena Mirkovic for delivering new capabilities under that program; to Bob Braden for careful crafting of proposals, capturing all of the prior work and proposed new work; and to John Sebes amanuensis extraordinaire.

7. REFERENCES

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

[2] The DETER Project - Advancing the Science of Cyber Security Experimentation and Test. Terry Benzel, Jelena Mirkovic, et al. *IEEE HST 2010 Conf*, Boston, MA, November 2010

[3] Vulnerability Detection Systems: Think Cyborg, Not Robot, S. Heelan. In IEEE Security and Privacy, special issue "The Science of Security," Vol. 9, Issue 3, May/June 2011.

[4] Justification and Requirements for a National DDoS Defense Technology Evaluation Facility, W. Hardaker, D. Kindred, R. Ostrenga, D. Sterne, R. Thomas, Network Associates Laboratories Report, July 26, 2002.

[5] Cyber defense technology networking and evaluation, R. Bajcsy, T. Benzel, M. Bishop, B. Braden, C. Brodley, S. Fahmy, S. Floyd, W. Hardaker, A. Joseph, G. Kesidis, K. Levitt, B. Lindell, P. Liu, D. Miller, R. Mundy, C. Neuman, R. Ostrenga, V. Paxson, P. Porras, C. Rosenberg, J. D. Tygar, S. Sastry, D. Sterne, S. F. Wu. In Communications of the ACM, Special issue on "Emerging Technologies for Homeland Security," Vol. 47, Issue 3, pp 58-61, March 2004.

[6] Preliminary results using scale-down to explore worm dynamics, Nicholas Weaver, Ihab Hamadeh, George Kesidis and Vern Paxson. In Proceedings of the 2004 ACM workshop on Rapid Malcode, pp. 65-72, 2004.

[7] A hybrid quarantine defense, P. Porras, L. Biesemeister, K. Levitt, J. Rowe, K. Skinner, A. Ting, In Proceedings of ACM WORM, Washington, DC, Oct. 29, 2004.

[8] Combining visual and automated data mining for near-real-time anomaly detection and analysis in BGP, S.T. Teoh, K. Zhang, S.-M. Tseng, K.-L. Ma and S. F. Wu, In Proceedings of ACM VizSEC/CMSEC-04, Washington, DC, Oct. 29, 2004.

[9] T. Faber and J. Wroclawski, "A Federated Experiment Environment for Emulab-based Testbeds," in Proceedings of Tridentcom, 2009.

[10] Stephen Schwab, Brett Wilson, Calvin Ko, and Alefiya Hussain, "SEER: A Security Experimentation EnviRonment for DETER," in Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test, August 2007.

[11] Emulab Testbed Web page, http://www.emulab.net

[12] https://trac.deterlab.net/wiki/Topologies

[13] DDoS Experiment Methodology, Alefiya Hussain, Stephen Schwab, Roshan Thomas, Sonia Fahmy and Jelena Mirkovic. In *Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test*, June 2006

[14] *An Integrated Experimental Environment for Distributed Systems and Networks*, by White, Lepreau, Stoller, Ricci, Guruprasad, Newbold, Hibler, Barb, and Joglekar, appeared at OSDI 2002, December 2002

[15] Operating System Support for Planetary-Scale Network Services. A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak. (*NSDI '04),* May 2004

[16] J. Wroclawski, J. Mirkovic, T. Faber and S. Schwab, "A Two-Constraint Approach to Risky Cybersecurity Experiment Management," Invited paper at the Sarnoff Symposium, April 2008.

[17] Barford, Paul; Landweber, Larry. *Bench-style Network Research in an Internet Instance Laboratory*, In Proceedings of SPIE ITCom, Boston, MA, August, 2002

[18] A Plan for Malware Containment in the DETER Testbed Ron Ostrenga and Stephen Schwab, Robert Braden. In *Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test*, August 2007

[19] T BitBlaze: A New Approach to Computer Security via Binary Analysis. Dawn Song, David Brumley, Heng Yin, Juan Caballero, Ivan Jager, Min Gyung Kang, Zhenkai Liang, James Newsome, Pongsin Poosankam, and Prateek Saxena. In *Proceedings of the 4th International Conference on Information Systems Security, Keynote Invited Paper.* December 2008

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

[21] Ted Faber, Mike Ryan, and John Wroclawski "Building Apparatus for Multi-resolution Networking Experiments Using Containers", in submission

[22] **"Scalability and Accuracy in a Large-Scale Network Emulator,"** Amin Vahdat, Ken Yocum, Kevin Walsh, Priya Mahadevan, Dejan Kostic, Jeff Chase, and David Becker. *Proceedings of 5th Symposium on Operating Systems Design and Implementation (OSDI)*, December 2002

[23] TSilva, Vladimir (11 March 2009). *Practical Eclipse Rich Client Platform Projects* (1st ed.). Apress. p. 352. ISBN 1430218274.

[24] Clifford Neuman and Kymie Tan, "Mediating Cyber and Physical Threat Propagation in Secure Smart Grid Architectures", To Appear in the Proceedings of the 2nd International Conference on Smart Grid Communications (IEEE SmartGridComm), October 2011, Brussels