ISI-TR-2006-613
March 2006

EXPERIENCE WITH DETER: A TESTBED FOR SECURITY RESEARCH

Terry Benzel, Robert Braden,
Dongho Kim, Clifford Neuman
Information Sciences Institute / University of Southern California

Anthony Joseph and Keith Sklower
Department of Computer Science
University of California, Berkeley

Ron Ostrenga and Stephen Schwab
SPARTA, Inc. Information Systems Security

ISI-TR-2006-613

March 2006

University of Southern California
Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292-6695
(310) 822-1511

EXPERIENCE WITH DETER: A TESTBED FOR SECURITY RESEARCH

Terry Benzel, Robert Braden,
Dongho Kim, Clifford Neuman
Information Sciences Institute
University of Southern California

Anthony Joseph and Keith Sklower
Department of Computer Science
University of California, Berkeley

Ron Ostrenga and Stephen Schwab
SPARTA, Inc.
Information Systems Security

ABSTRACT

The DETER testbed is shared infrastructure designed for medium-scale repeatable experiments in computer security, especially those experiments that involve malicious code. The testbed provides unique resources and a focus of activity for an open community of academic, industry, and government researchers working toward better defenses against malicious attacks on our networking infrastructure, especially critical infrastructure. This paper presents our experience with the deployment and operation of the testbed, highlights some of the research conducted on the testbed, and discusses our plans for continued development, expansion, and replication of the testbed facility.

1. INTRODUCTION

Rapid advances are urgently needed to defend against network attacks such as distributed denial of service, worms, and viruses. These cyber-security problems include some of strategic importance, like the protection of critical infrastructure. Rapid advances require an improvement in the state of the art of experimental evaluation of network security mechanisms.

Such efforts require the development of large-scale security testbeds [6], combined with new frameworks and standards for testing and benchmarking to make the testbeds truly useful. Current impediments to evaluating network security mechanisms include lack of scientific rigor [15]; lack of relevant and representative network data [13]; inadequate models of defense mechanisms; and inadequate models of the network, background, and attack traffic data [4]. The latter is challenging because of the complexity of interactions among traffic, topology, and protocols [4,5].

Cyber-defense research has been severely limited by the lack of a public experimental infrastructure for testing new theories and new technologies in realistic scenarios. It is both unclear and unproven that technologies tested on small subnet-sized topologies modeled by a few machines will scale to realistic Internet environments.

To meet this challenge, the cyber-DEfense Technology Experimental Research (DETER) testbed [2] has been developed. The DETER testbed is intended to provide an experimental infrastructure to support the development and demonstration of next-generation information security technologies. DETER provides a medium-scale facility for safe, repeatable security-related experimentation, to validate theory and simulation. The DETER testbed is implemented as an Emulab [21] cluster, using the comprehensive and powerful cluster testbed control package developed by Jay Lepreau and his colleagues at the University of Utah.

With a current design point of several hundred experimental nodes, the DETER testbed provides an intermediate point between small-scale and Internet-scale experiments. Since it is chartered to support scientific investigation, the testbed is designed with experimental repeatability as a fundamental requirement. Repeatability allows experimenters to deeply investigate, validate, and find alternative explanations for their research results and to build upon the results of others.

In addition to the hardware and software infrastructure needed to conduct experiments, the DETER testbed provides tools that aid the experimenters, many of which are being developed by experimenters themselves. These experiment support facilities are introduced briefly in Section 4.

The public Internet must be protected from the side effects of the security experiments that run on the testbed and the experiments must be protected from interference from the Internet. The DETER testbed must provide containment of malicious code as well as control over the effects of generated attack and background traffic. Because different experiments pose different levels of threat to the public Internet, it is important to balance on a case-by-case basis the cost and complication added by isolation with the level of threat posed by the individual experiment.

Terry Benzel, Bob Braden, Dongho Kim, Cliford Neuman. Anthony Joseph and Keith Sklower Ron Ostrenga and Stephen Schwab, Experience with DETER: A Testbed for Security Research. In Proceedings of the 2nd IEEE Conference on testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom2006). Barcelona, Spain. IEEE. March 2006.

This paper presents our experience with the deployment and operation of the testbed, highlights selected projects, and discusses our plans for continued development, and expansion of the testbed facility. Section 2 summarizes the requirements and the design of the DETER testbed. Section 3 discusses the central security issues and the measures taken to counter the threats. Section 4 discusses the experiment support facilities that are being developed by and for the DETER research community. Section 5 presents some important examples of experimental applications of the testbed. The paper finishes with Lessons Learned in Section 6, and Conclusion in Section 7.

2. OVERVIEW OF TESTBED DESIGN

While the hardware is an important feature of the testbed capability, testbeds generally require very significant control software, to support rapid reconfiguration of the testbed, and secure partitioned time and space sharing of the testbed for multiple users. The generality, flexibility, and usability of the control software is critical for a testbed to meet the needs of its users. The expense of developing and maintaining the software can easily exceed the hardware cost, sometimes substantially. Fortunately, the Utah Emulab software was available to meet most of our needs.

2.1. Testbed Requrements

A simple testbed can be constructed by manually wiring together and configuring a dedicated set of machines; however, such a testbed lacks generality and sharability. DETER (like Emulab) belongs to the more useful class of testbeds that are *general-purpose* and support *remote access*. It can be used effectively for a wide variety of experiments, and an experimenter can reconfigure and control experiments remotely. Additionally, DETER is *partitionable* into multiple independent experimental testbeds that can be used simultaneously, allowing more efficient use of its hardware resources.

For all but the most dangerous experiments, the testbed must be remotely accessible to experimenters for initiation and monitoring. An experimenter must be able to control the experiment even when the test network is congested or broken as the result of the experiments that are run. The requirement for remote accessibility may clash with security and containment requirements.

The testbed must also be sharable in time and space among a large community of users, while providing strong isolation of effects between users (e.g. traffic, attacks, etc). Just as a major particle accelerator needs to have multiple beam-lines, so the DETER testbed needs to support multiple simultaneous experiments.

These properties of general-purpose, remote access, and partitionabililty, imply at least the following classes of functionality in the control software: [21]

- User accounts and access control
- Allocation of hardware resources to each partition
- Automatic configuration of hardware and software within a partition
- Remote user access and control during experiments

Efficient operation is critical for successful time and space sharing. For example, the control software must be able to install system images on hundreds of nodes in parallel. [8]

The basic requirements for the DETER testbed include general purpose, remote access, and partitionabililty. Other important requirements are containment and security, fidelity, repeatability and reproducibility, programmability, and scalability. Containment and security are discussed in Section 3. The other requirements are discussed below.

2.1.1. Fidelity

Fidelity to “real” networks, and in particular to the real Internet, is important. There are several dimensions to fidelity: (1) the number of nodes, (2) realism, i.e., reproducing real router and end-system behavior, and (3) realistic heterogeneity of hardware and software, and (4) a realistic mix of link bandwidths and delays.

Fidelity has costs for the purchase, maintenance, and operation of hardware and software. The hardware-related costs of a testbed increase linearly with the number of nodes, and faster than linearly when the cost of switches is considered since some switch ports have to be used for inter-switch bandwidth. A central aspect of the experimental science on testbeds is to construct idealized abstractions of the real Internet with *enough* fidelity for specific experiments.

Some experimenters will want to run experiments that require more nodes than are available. It is possible to run multiple virtual nodes on each physical node to enable such experiments, but virtualization introduces artifacts which must be considered when evaluating experimental results.

Some experimenters will want to run experiments using real hardware that is not regularly part of the DETER testbed. These users need to work with the actual software of common commercial routers or to perform live tests on commercial security hardware appliances. We are extending Emulab software by adding several commercial routers and appliances to the testbed.

2.1.2. Repeatability and Reproducibility

A critical objective of DETER is to support scientific experiments that should be repeatable. The dynamics of the real Internet cover a wide range of conditions and measurements on the real Internet vary widely from day to day and place to place. The size, available bandwidth, versions of software in use, and the background “attacks” and user traffic that are present are continually evolving. As such, by the time a result is published, the Internet has changed irrecoverably, making it impossible to truly repeat an Internet experiment.

An important objective of the testbed, on the other hand, is to allow the experimenter to repeat experimental conditions precisely (from a statistical standpoint), modifying them only in a controlled manner. By implementing the testbed as a large set of experimental nodes, we are able to provide more uniform control over the environment. The testbed control software enables the collection and storage of complete disk images in a database, and can be used to capture the operating system configuration, program, input, and output state of an experiment. Storing disk images is not efficient for recording program inputs and outputs, so we are collecting and archiving logs and output files for subsequent analysis and we plan to develop separate database mechanisms for their storage.

Repeatability also depends on having tools that allow experimenters to measure the statistical variability of metrics of interest, and to record the variances for later verification of subsequent experiments. We are developing tools to enable such data measurement and collection.

Repeatability includes a requirement for the absence of testbed artifacts caused by concurrent experiments. Once we have support for recording variance metrics, we will have full support for reproducibility. Other researchers will be able to reproduce experiments from the testbed at a later time, *and* confirm that the metrics from re-running the experiments match the statistical variability of the original experiment’s metrics.

2.1.3. Programmaility

Emulab uses high-performance VLAN-capable switches to dynamically create nearly arbitrary topologies among the nodes. However, a significant set of DETER experiments involve testing new monitoring, filtering, and diagnosis mechanisms within the network. This implies adding or modifying router algorithms. Router vendors are not anxious to open their platforms to experimental modifications, so we support a special node image for PC-based software routers, such as the Click and Zebra routers [14,23]. This node image has multiple routable interfaces and can run experimental routing protocols. Using software routers adds flexibility and programmability with some sacrifice of fidelity.

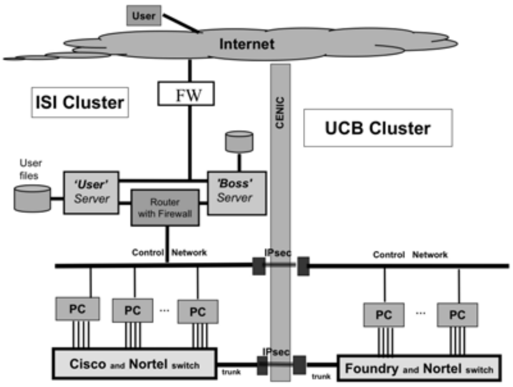

Fig. 1. Architecture of the Testbed.

2.1.4. Scalability

The Emulab code provides the Deter testbed with methods for assigning, initializing, and managing multiple experiments. We have improved the scalability of the DETER testbed by extending the Emulab [21], software to support two clusters, one at USC ISI and the other at UC Berkeley. The Emulab control software for DETER is configured to place nodes at the two sites in separate logical pools. An experiment can allocate nodes from either one or from both clusters.

2.2. Testbed Design

Figure 1 shows a simplified view of the DETER testbed architecture, based upon Emulab [21]. It shows the two clusters of experimental PC nodes, at ISI and at UC Berkeley. These clusters currently contain roughly 100 nodes each. These nodes are interconnected by a "programmable backplane" of high-speed Ethernet switches, trunked to form a single logical switch. Each PC has four interfaces to this switch. The Emulab control software, running on the 'Boss' server, allocates available PC nodes to an experiment and interconnects them by setting up VLANs in the switches, to create the topology specified by the experimenter. The Ethernet links are 10/100/1000 Mbps. High-capacity switch hardware is necessary to avoid experimental artifacts caused by interference between VLANs.

In addition to the four data plane interfaces, each experimental PC node has an interface on the shared Control Network. This network is used by the Emulab software to download, configure, and monitor the nodes, and may be used by the experimenter to control and monitor the nodes. Finally, for simplicity Figure 1 omits two other paths to each experimental node: a serial line server connected to all the console ports, and a programmable power controller that allows power cycling of the nodes individually.

The 'Boss' server provides a comprehensive GUI-based user interface to the Emulab control mechanism, and it controls the switches and power controllers. The 'User' server has accounts for experimenters. A running experiment can NFS-mount the user's directories. These accounts can also be used by experimenters to access experimental nodes, using relayed SSH. This provides remote access to the experimental without creating a direct IP path to the general Internet.

The trunking connections between the two clusters use the Califonira High-Performance Research Network, a high-speed (1Gbps) research network that connects Los Angeles to Berkeley. To preserve integrity and privacy, this interconnection uses IPsec encryption. There are separate paths for the data plane and for the Control Network, as shown in the figure.

As Figure 1 shows, the experimental plane is distributed between ISI and UC Berkeley but the testbed is controlled from ISI. This centralization of control and administration is desirable from the viewpoint of a user, who wants a single point of access to the combined facility. In also creates a possible single point of failure. It has proven useful to have a secondary 'User' and 'Boss' server at UCB (not shown). A manual reconfiguration of the Emulab control tables transfers a pool of UCB nodes between ISI control and UCB control. This has been useful for testing new versions of the Emulab software in an isolated environment. It could also be exploited to provide a 'hot' backup in case of failure of the control components at ISI.

3. SECURITY ISSUES

Security is not only the object of research using the testbed; it is also a vital requirement for the testbed itself. Security for the DETER testbed is critical, and the threats are both internal and external. Internal threats come from virulent code that is tested within DETER and threatens to take control of the testbed or escape into the Internet. Additional internal threats can come from experimenters who attempt to steal test data or results from other investigators, prior to publication. The external threats come from hackers who see the testbed as a tempting target for an exploit; thus, infiltration protection is required.

Like any network infrastructure connected to the Internet, the DETER testbed is subject to attack; this is especially acute because a security testbed forms an attractive target. Both experiments running in DETER and the testbed control plane must be protected. Because of DETER's mission, DETER security has a major extra component: the public Internet as well as the testbed control plane and other experiments must be protected from attack by experiments running in DETER. Whereas security in most systems is concerned only with the problem of *infiltration*, DETER is additionally concerned with the problem of *exfiltration*.

The intent of the DETER testbed is to provide containment for security experiments, to support safe experiments that present a wide range of threat levels. The most dangerous level might be “live” testing of a contagious attack program whose attributes are completely unknown, for example an actual malicious worm or virus. The traditional approach to testing such dangerous programs has used a completely isolated laboratory consisting of dedicated systems whose disk drives and memory chips never leave the laboratory and are never reused. Experimenters must be physically present in these laboratories and must be specially trained.

The approach of the DETER project, on the other hand, is to build a single safe testbed that can change its operational mode to match the threat level of the experiments. DETER provides a shared laboratory facility for those experiments whose threat level is low enough to allow sharing, but it can be reconfigured for exclusive use for more dangerous experiments. The testbed allows remote experimenter access for all but the most dangerous experiments.

Figure 1 shows that the Control Network provides a shared path that could allow an experiment to attack another experiment or the 'Users' or 'Boss' servers. This shared path is a major security issue with the basic Emulab design. Utah has recently enhanced Emulab with new features to ameliorate this vulnerability. The DETER project is in the process of installing and evaluating the full security implications of these improvements.

This section briefly discusses security requirements for the DETER testbed and explains how the requirements are met.

3.1. Containment

Containment addresses the need to prevent exfiltration of packets from the testbed. The worst breach of containment would be release of a previously unseen virus or worm into the public Internet. In addition to containing malicious code, the testbed contains the effects of malicious software and excessive traffic that are generated by an experiment.

The DETER testbed provides containment through several means. The first is the use of a physically separate experimental network on which the nodes of an experiment communicate. This network is unable to route packets beyond the nodes that are part of the experiment. Second, as shown in figure 1, firewalls are placed at several locations in the testbed and on the interface between the user machine and the open Internet

Additionally, it is easy to detect traffic on the control network other than that which is expected, such as that caused by misconfiguration of an experimental node which sends experimental traffic to the control network. This raised an alert regarding problems with the experimental or control nodes for an experiment.

The problem of containment is a little more complicated when considering dissemination of malicious code that has been running within an experiment. While active exfiltration can be prevented by the techniques just described (i.e. no route for packets to leave the testbed), malicious code can escape by hiding itself in data retrieved to the outside by an experimenter upon conclusion of an experiment. At present we are not concerned with an experimenter intentionally retrieving such code – that is left to management controls. Although malicious code has not yet been written with specific knowledge of DETER and how to hide itself, it is a concern down the road. Our planned approach is to require data retrieved from the testbed to be encrypted and encapsulated in a manner that prevents unintentional execution by the receiver.

3.2. Isolation

We provide physical link isolation to address the need to prevent experiments from interfering with one another, or for events external to the experiment or the testbed to interfere with the results of an experiment. Such interference could be unintentional, such as the overloading of a common network link by another experiment, or intentional such as from of a denial of service attack.

A programmable VLAN switch is used to map physical connections between nodes, so there is effectively no interference between links of the same or different experiments, as long as the nodes are allocated on the same switch. Because multiple switches are needed to handle the total number of nodes in the testbed, as well as to handle nodes at different physical sites, the testbed contains multiple switches connected by trunking links, some of which are used within a site, and some of which are wide area. These links may be over-subscribed by the logical links crossing them, which can cause experimental artifacts (varying performance) when a single experiment uses nodes on multiple switches. Careful monitoring of connection bandwidth is used to alert investigators if interference (or even just plain over allocation) has occurred. Finally, interference on the control network is limited to the setup, breakdown, and monitoring of an experiment and would not affect the results unless stray packets are sent to an experimental node, but such packets can be detected and the investigator alerted to the problem.

3.3. Confidentiality and Integrity

The confidentiality requirements of the DETER network center around the protection of the data used by an experiment, the code and nature of the experiment, and the results of the experiment until such time as the results are published. Often an experiment will use input data such as traffic traces that are subject to non-disclosure agreements,

Confidentiality must be provided while data is resident in the staging area for an experiment (databases and file systems), in place on nodes assigned to an experiment, and while transiting the network. Confidentiality of the data while resident on the staging file system, and while it is in transit to an allocated experimental node is provided through the use of the Cryptographic File System [1].

Additional confidentiality is achieved by overwriting (zeroing) node disks when an experimental node is deallocated from an experiment, and before a new system image is loaded for the subsequent experiment

Integrity of the data used as inputs to and produced by experiments is also critical. The integrity issue is addressed through the use of a cryptographic file system.

3.4. Achieving Security

The security goals just described are achieved in the DETER testbed through several techniques, many of which are not unique to DETER. These techniques include:

3.4.1. Firewalls

Firewalls are deployed both externally and internally in the DETER architecture, as shown in Figure 1. The internal firewall is configured to help protect the control plane from disruptions by experiments and to prevent exfiltration of experiments. These egress filters are redundant because the topology of the testbed itself prevents such egress, but they do provide protection in case someone plugs a connector into an incorrect port.

Firewalls are also deployed extensively on the control network. Because the expected communicating pairs of machines on the control network is well constrained and the ports and protocols for such communication are known in advance, the configuration of these firewalls is very restrictive.

3.4.2. Intrusion detection

The placement of an intrusion detection system on the network between the staging machine and the Internet, and the planned addition of an intrusion detection system on the control network will allow us to detect traffic that should not be there. Again, because the means of interaction with experiments is well constrained, we can write relatively tight rules for anomaly detection, that have a very low rate of false positives.

We are also currently exploring the possibility of deploying additional intrusion detection mechanisms to detect anomalous behavior within experiments. While activity on the control network and interface to the outside is very constrained, fewer constraints exist on activity within the experiments themselves.

To aid in detecting misbehaving experiments, we can require investigators to provide us with a characterization of the behavior of their experiment when they propose to use DETER. This characterization can be used both to determine the level of containment needed for their experiment, and to load rules into an intrusion detection system. Experiments that exceed their proposed envelope by certain bounds would be immediately suspended. If the problem was poor characterization by the experimenter, they can update their request for access and proceed, possibly under a new set of containment rules.

3.4.3. Decontaminating Nodes

Upon deallocation of an experimental node, the disks on the node are zeroed and a new system image loaded for the subsequent experiment. This protects confidentiality of the data that was resident on the node and prevents interference with the next experiment, which could occur if changes were made to the system image.

3.4.4. Red Teaming

To help verify the security of the DETER testbed, a red team from Sandia Laboratories was contracted for a security assessment. Several exploitable vulnerabilities were found, some specific to the DETER configuration and address translation. As a result of this test, configuration changes were immediately made to the DETER testbed addressing the vulnerabilities.

3.4.5. Administration

The last technique used for protection of the DETER testbed is administrative rather than technical. Investigators seeking to use DETER must submit an application which is classified according to the potential threat the experiment poses to the testbed and to the Internet in general.

The investigator is asked to describe the potential threats resulting from a breach of containment, and explain the basis for that assessment (for example, to tell us why a simulated worm could not affect computers running beyond his or her experiment). The investigator is also asked for any confidentially or other specific security requirements.

We review proposals to assess the threat posed to the public Internet by breach of containment. We take a conservative view in our assessment of the investigators statements, considering possible errors that can be made.

Once a proposal to use the testbed is approved, the safety committee determines the level of containment required, and this might affect whether the experiment can run concurrently with the experiments of other users, and whether remote access to the control side of the experiment is to be supported. For particularly dangerous experiments, the constraints could require running the testbed in an isolated mode of operation, while for most experiments, the testbed's isolation mechanisms are sufficient.

3.5. Protection Domains and Federation

The DETER testbed is presently managed as a single entity, even though nodes are present both at USC and at UC Berkeley. Common policies are applied at both locations, and the interconnection between sites is accomplished through two encrypted tunnels, one on the control network and one on the experimental network.

One of the experiments run on DETER did not fit this model and required special hardware connected to the testbed through an encrypted tunnel to the user's site. In this particular case we had confidence in the investigator and allowed the departure from our normal configuration. The placement of the tunnel was such that a breach of security at the remote hardware would affect only this one experiment, although that level of containment was dependent on the proper functioning of our other defenses. We are investigating a generalization of this capability, to provide a protected *portal* from the testbed for access to special hardware running in a protected laboratory external to the DETER testbed itself. Such a model is important for support of industrial users with new types of hardware routers and security appliances.

We are also investigating capabilities for federation of multiple testbeds. As independent DETER-like clusters are established, there is a need for interconnection of such clusters to enable the running of larger experiments. For such systems, the security model of a single management domain will no longer apply and more complicated policy issues will need to be addressed.

4. EXPERIMENT SUPPORT FACILITIES

The DETER testbed is aimed at a relatively narrow experimental community, so it provides an opportunity to create and maintain a set of common software tools and data that support security experiments. The testbed builders and the testbed users are collaborating to build such a common set, including building blocks, a repository of complete experiments, and an integrated experimental environment.

Testbed operations has been assembling a library of useful building blocks: tools and data sets. The tools include generators for sample topologies, for network configurations, and for attack and background traffic. They also include a variety of measurement, data analysis, and visualization tools. Many of these tools are being developed by DETER experimenters to ease their own work. The data sets will contain standard topologies and other static information that will facilitate comparative experiments using common network conditions.

The repository of complete experiments will contain both very simple experiments, to help newcomers and students, and also some paradigmatic complex experiments, to serve as a basis for modification to build new experiments.

Finally, we intend to integrate these tools and facilities into a common experimental environment, to ease the task of creating and running a complex experiment. We refer to the goal of this effort as the "security experimenters' workbench" (SEW). The SEW will aid the assembly of a complete experiment, including topologies, generators, configurations, and monitoring tools. Built around a GUI, it will provide an organized interface for monitoring and controlling execution of the resulting experiment, and it will provide a powerful set of tools for analyzing the results. Finally, the SEW should facilitate repeating an experiment with different parameters and algorithms. An initial model SEW, called ESVT (Experiment Specification and Visualization Tool) has been very successful [10]. Future work is expected to include the specification and development of common interface standards for experimental tools.

5. THE EXPERIMENTAL PROGRAM

The DETER testbed has been live since March 2004. To illustrate the strength as well as the limitations of the DETER testbed for security research, this section briefly describes several research programs using DETER. We discuss DDoS, worm behavior, and BGP security experiments, as these cover a range of technical demands.

These research efforts have been aimed at experimental verification of the effectiveness and dynamics of attacks and defenses, and also at the development of methodology of these experiments. The methodology research has led to the development of libraries and tools that have been made available to other users of the DETER testbed. This work has been captured in a set of prototypical experiments (benchmarks) and associated databases of:

- topologies and topology generators
- attack and background traffic traces and generators
- defenses
- special-purpose devices (meters, virtual nodes, etc.)
- metrics for scale-down, fidelity, performance, overhead.

Many of the experiments described here were performed by researchers from the EMIST [2] project, which was funded concurrently with DETER. EMIST is a collaboration among UC Davis, Penn State, Purdue University, SRI International, ICSI, and SPARTA. The DETER user community has since expanded to include over a dozen researchers from academia and industry.

5.1. DDoS Experiments on DETER

DDoS experiments on DETER have explored the dynamics and effects of DDoS attacks on complex networks. They contributed to the development of a methodological framework for analyzing the effectiveness of DDoS defense technologies [7]. This framework was refined through experiments of increasing scale and realism, using combinations of simulation, emulation on the DETER testbed, modeling, and analysis. A notational short hand was developed for describing and comparing experiments, archiving experiment descriptions, data, and results. [17]. This archive will be expanded to cover other defensive technologies and attack scenarios, and will serve as a set of resources for other DDoS experimenters, making it relatively easy for new experimenters to reuse existing software and tools to create an experiment scenario.

DDoS experiments on DETER have included:

- studies of defensive technologies, using commercial and open source software, and research prototypes;
- investigation of configuration, conduct, methodology and analysis of DDoS defense, in a rigorous setting;
- examination of two specific commercial software packages Symantec ManHunt and Network Flight Recorder (NFR) Sentivist.
- evaluation of FloodWatch [3], a traffic detection and response system using statistical profiling, as a defensive technology for defining, executing, and refining the experimental process.

The initial set of DDoS experiments was aimed at understanding the ability of defensive mechanisms to differentiate background traffic from attack traffic. Key observations of the experiments included the generation of false alarms and understanding the phenomenology of attacks. For the latter, experimenters observed the effect of the choice of attack types and parameters in order to understand first-order and second-order attack effects. Later experiments validated the fidelity of attack and background traffic in reproducing characteristics of real DDoS experimental scenarios**.**

Having established experimental testbed requirements and created baseline testbed mechanisms, the DDOS team is now studying scientifically-based testing methodologies. This research defines canonical forms, provides resources, refines a process for comparable experiments, enhances experiment automation, provides archival capabilities for analysis, and identifies limitations on scale and realism. Drawing on this experience with the DETER testbed the DDoS Requirements are being generalized to broader security experiment requirements and the development of automated support for repeatability, efficiency, and ease-of-use that will assist future researchers on the testbed.

5.2. Worm Behavior Experiments using DETER

Early DETER experimentation on worm behavior concentrated on modeling Internet-scale dynamics of worm propagation. Since it is not possible to perform a truly Internet-scale experiment, scale-down is a critical for experiments on worm behavior.

Early worm behavior research on DETER included:

- Development of two models for scanning worms [20]: the homogeneous cluster model and the heterogeneous cluster model. These models were simulated and executed on DETER using a 1/64 scale-down factor, enabling real time emulation Future efforts will provide a better scale-down with fewer artifacts and consider varying capacities of the access links, not just the distribution of infected machines.

- A 1000 (virtual) node enterprise network simulations of the Slammer worm [11], and Witty and Blaster worms [12]: This included development of virtual nodes that model the response of subnetworks to a worm attack for the purposes of studying scale-down. This experiment also employed a visualization tool that has been integrated into the DETER testbed for use by other researchers [10].

- Development of an abstract data model called the Internet Worm Propagation Data Model (IWPDM) [9] and WormGen, a safe attack generation system. In order to expand the test to a larger number of nodes, each physical node on the DETER testbed hosted four WormGen agents, using the four network interfaces on each physical node. Results measured infection rates for a random scanning worm and a topological scanning worm.

A number of experimenters using the DETER testbed have reported significant results through the use of the testbed. A researcher from ICSI reported recently that he discovered a problem in his worm emulation model as a result of running source models and emulated worms on the testbed.

Researchers from SRI and UC Davis report that the emulated DETER environment can support execution of real worm code and defense algorithms, providing deeper insight into attack defense dynamics than can be obtained through simulation [16].

A third researcher discussed the role that fidelity plays in the accuracy of models [18]. It is expected that an increase in model accuracy resulting from the addition of nodes, and reduction of testbed artifacts will lead to increased understanding of worm behavior and thus to opportunities for new research on worm containment and prevention techniques.

5.3. Early Routing Security Experiments in DETER

Preliminary research on BGP routing attacks [19,22] has:

- Evaluated the ability of security mechanisms such as Whisper/Listen, SBGP, and SoBGP to defend the Internet routing infrastructure against malicious attack.
- Demonstrated two types of BGP attacks: OASC (Origin AS Changes) and DDP (Differential Damping Penalty). The experiments provided data that could be used to compare their strength, weakness, performance, and the effectiveness of several proposed approaches to handle attacks toward the routing infrastructure.
- Examined signature- and statistics-based detection to search for anomalous BGP routing dynamics.

The experimenters proposed two approaches to managing this analysis and are conducting experiments to identify advantages and limitations of each. Their study is currently limited by the lack of data from real environments, but the use of the DETER testbed supports examination of BGP on a larger scale.

6. LESSONS LEARNED

We have learned that the needs of users vary more significantly than originally expected. Security experiments tend to be larger than other experiments because the effects of attacks are often not felt until a large number of end machines have been compromised. Some experiments require large numbers of nodes, many more than we can provide physically, so even with 300 nodes, support for virtual nodes was important. The use of virtual nodes, however, introduces artifacts in the experimental results that must be considered.

We also found that some experiments required the ability to employ topologies of specific commercial routers which had not been previously incorporated into DETER. This introduced new requirements for testbed design to enable investigators to plug in hardware modules, in this case, separately from the testbed and interconnected through an encrypted tunnel, and to allow those modules to be allocated only for specific experiments.

We believe that the portal concept design can be extended to other inter-testbed connections and integrated into the DETER control plane. However, it should be noted that such connections require the use of dedicated nodes, thereby reducing the total number of nodes available for experimentation. If the testbed is to support more portals then more dedicated nodes will be needed for this use.

Many of the experiments that run on DETER do not require the most secure mode of containment. Basic containment is needed to keep the effects of an error in the traffic and attack generators from causing problems beyond the confines of the testbed, but because the code to be tested is written by the investigators, the incentive for breach of containment is just not there. However, the ability to experiment with wild malicious code is important for some experimenters, and the red teaming experiments were helpful in moving us toward that ability. The ability to run large scale simulations and emulations employing a mix of physical and virtual topologies can provide a mechanism for exploring “what if” scenarios and evaluating response options in the face of critical infrastructure attacks.

Common instrumentation and measurements are important issues for a testbed. While the general nature of the testbed nodes makes it possible for investigators to implement their own data collection and measurement tools, a common set of tools both eases the burden on new experimenters and ensures that results are comparable between experiments by different investigators. Furthermore, defining standards for the data streams containing measurements makes it easier to build a common platform or workbench through which investigators can select the data generators and measurement tools needed for their experiment, and/or to write their own when they need unique tools, but which will integrate readily with other parts of their experiments.

7. CONCLUSION

The DETER testbed has been operational since March of 2004 and is used by researchers to perform experiments on worm propagation, distributed denial of service attacks, and routing and infrastructure attacks. At the time of writing, the testbed had 231 nodes and it has been used by commercial and academic researchers to study attacks and assess the benefit of products in development.

The testbed provides investigators with the ability to run experiments using potentially risky code, on an isolated experimental network. For most categories of experiments, control is possible remotely by connecting to a testbed user machine through the Internet. Firewalls, intrusion detection systems to monitor access, and other safeguards protect access through this control network, and physical separation from the Internet is provided on the experimental network on which the experimental nodes communicate.

The testbed provides a focus of activity for a community of academic, industry, and government researchers. Regular meetings of the user community provides an opportunity for investigators to show early results, and to help one another in the use of the DETER testbed.

Support for testbed users includes a repository of attack traffic generators, monitoring tools, topology generators, and other tools, and work is underway to integrate these tools into an experimenters’ workbench which will simplify the task of getting new experiments up and running.

Over time we expect to see replicas of the DETER testbed, collectively supporting a larger user community, with varying requirements for containment, confidentiality, the number of nodes, and performance. We will explore ways to federate such independently managed testbeds to enable larger experiments to run than can be supported on a single testbed. We have experience running the DETER cluster across two sites, USC and Berkeley, however, we have managed these sites as a single domain with common security requirements. Federation of testbeds is a much more complicated problem which must take into account differences in the policies enforced at the different endpoints, differing levels of containment, diversity in the user communities and the level of trust one places in the management of the independent clusters.

The DETER testbed provides a venue for investigators to run experiments that require containment from release to the open Internet. The testbed provides an environment that makes experiments more readily repeated and validated by others, and serves as a repository for the data and hardware and software configurations used for experiments. For more information please visit http://www.isi.edu/deter.

8. ACKNOWLEDGEMENTS

This research was supported by funding from the United States National Science Foundation and the United States Department of Homeland Security under contract numbers ANI-0335298 (DETER) and CNS-0454381 (DECCOR). Opinions, findings, conclusions and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF). Figures and descriptions are provided by the authors and are used with permission.

Juniper Networks and Hewlett-Packard donated equipment used by the DETER testbed. Donations were also received from Sun Microsystems and Dell through their University Discount programs.

9. REFERENCES

[1] Blaze, Matt, A Cryptographic File System for UNIX, *ACM Conference on Computer and Communications Security*, pp. 9-16, 1993.

[2] Members of the Deter and EMIST Team, Cyber Defense Technology Networking and Evaluation, Communications of the ACM 47(3), March 2004, pp 58-61.

[3] Feinstein, L. et al. *Statistical Approaches to DDoS Attack Detection and Response*. In Proceedings of the Third DARPA Information Survivability Conference and Exhibition (DISCEX III). 1 (Apr. 2003) 303-314.

[4] Floyd, S. and Kohler, E. Internet research needs better models. *Hotnets-I* (Oct. 2002).

[5] Floyd, S. and Paxson, V. Difficulties in simulating the Internet. *IEEE/ACM Transactions on Networking 9*, 4 (Aug 2001), 392–403.

[6] Hardaker, W. et al. Justification and Requirements for a National DDoS Defense Technology Evaluation Facility. Network Associates Laboratories Report 02-052, July 26, 2002.

[7] A. Hussain, J. Heidemann, and C. Papadopoulos, "A Framework for Classifying Denial of Service Attacks," In Proceedings of *SIGCOMM 2003*

[8] Hibler, M., L Stoller, J. Lepreau, R. Ricci, and C. Barb. *Fast, Scalable Disk Imaging with Frisbee.* Usenix 2003.

[9] Levitt, K. et al. *Using a Worm Propagation Data Model for Safe Attack Generation Systems.* In Proceedings of CCS Workshop on Worm Behavior, ACM Conference on Computer and Communications Security. (Oct. 2004).

[10] L. Li, S. Jiwasurat, P. Liu, G. Kesidis, EMIST Experiment Specification and Visualization Tool: The User Manual, October, 2004. Code at: http://emist.ist.psu.edu/ESVT2/download_esvt2.html

[11] Li., L. et al. Worm Propagation in Enterprise Networks Emulated on the DETER Test-bed. *Technical Report, School fo IST and CSE Department, Penn State University* (June 2004).

[12] Li., L., I. Hamadeh, S. Jiwasurat, G. Kesidis, and P. Liu. Emulating sequential scanning worms on the DETER testbed. In the *2nd International IEEE Conference on Testbeds and Research Infrastrucutres for the Development of Networks and Communities,* Barcelona, (March 2006).

[13] McHugh, J. Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system valuations as performed by Lincoln Laboratory. *ACM Transactions on Information and System Security 3*, 4 (Nov. 2000), 262–294.

[14] Morris, Robert, Eddie Kohler, John Jannotti, M. Frans Kaashoek. The Click modular router, *Symposium on Operating Systems Principles*, pp 217-231. 1999.

[15] Pawlikowski, K., Jeong, H., and Lee, J. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine* (Jan. 2001).

[16] Porras, P., L. Briesemeister, K. Skinner, K. Levitt, J. Rowe, and Y. Ting. A Hybrid Quarantine Defense. In Proceedings of Worm'04 (October 2004).

[17] Schwab, S., B. Wilson, R. Thomas, "Methodologies and Metrics for the Testing and Analysis of Distributed Denial of Service Attacks and Defenses," MILCOM, Atlantic City, NJ, Oct. 2005.

[18] Sewani, A., "A System for Novel Email Virus and Worm Detection," Masters Report, University of California, Electrical Engineering and Computer Science department, August 2005.

[19] Teoh, S. et al. Combining Visual and Automated Data Mining for Near-Real-Time Anomaly Detection and Analysis in BGP. In Proceedings of CCS Workshop on Visualization and Data Mining for Computer Security, ACM Conference on Computer and Communications Security (Oct. 2004).

[20] Weaver, N. et al. Characterization, Modeling and Scale-down Simulation of Slammer's Propagation in the Internet. *In Proceedings of CCS Workshop on Worm Behavior, ACM Conference on Computer and Communications Security* (Oct. 2004).

[21] White, B., J. Lepreau, L.Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An Integrated experimental environment for distributed systems and networks. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI02),* (Dec. 2002). Pp 255-270.

[22] Zhang, K. et al. On Detection of Anomalous Routing Dynamics in BGP. *Networking 2004,* 259-270.

[23] The Zebra Router. http://www.zebra.org.