Very Large Scale Cooperative Experiments in Emulab-Derived Systems

Keith Sklower Anthony D. Joseph
University of California, Berkeley
{*sklower, adj*}*@cs.berkeley.edu*

Abstract

Cyber-defense research has been severely limited by the lack of an experimental infrastructure for testing new theories and new technologies in realistic scenarios. Current testbeds are mostly small-scale and limited to small numbers of machines. The cyber DEfense Technology Experimental Research (DETER) testbed, provides a medium-scale test environment with more than 300 nodes. However, there is increasing interest in running experiments at very large scale with more than 1,000 nodes.

This paper describes how experiments can be federated across existing small- and medium-scale testbeds using the University of Utah's Emulab software, such as the DETER testbed, to enable the running of massive-scale experiments. We describe the Emulab software and the DETER testbed and we detail the necessary steps for running a federated experiment. We provide a status update on our progress and discuss how a manually configured proof-of-concept experiment could be performed.

1 Introduction

Cyber-defense research has been severely limited by the lack of an 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 up to realistic Internet-scale environments. To perform detailed emulation and analysis of the behaviors of large systems under attack (e.g., the Internet or large enterprise networks), significant numbers of computers are required. As a step towards addressing this need, the cyber DEfense Technology Experimental Research (DETER) testbed [1, 2], which currently contains more than 300 nodes, provides an intermediate point in this scaling range that has turned out to be a very useful scale for many experiments.

The DETER testbed is open, free, shared infrastructure designed to support research and education in cybersecurity. The testbed supports medium-scale repeatable experiments in computer security, especially those experiments that involve malicious code or cannot be performed in the Internet because of traffic volumes or the risk of escape. The DETER testbed provides a unique experimentation facility where academic, industrial, and government researchers can safely analyze attacks and develop attack mitigation and confinement strategies for threats such as Distributed Denial of Service defenses, virus propagation, and routing security. In addition, the testbed provides tools and resources to enable repeatable scientific experiment methodologies, allowing researchers to validate their own theories, simulations, and emulations, while also enabling different researchers to repeatably duplicate and analyze the same experiments.

The DETER testbed is controlled by a version of Utah's Emulab software [3] configured and extended to provide stronger assurances for isolation and containment. With its strong security, containment, and usage policies, the testbed fills a role that is currently not met by any of the other large-scale testbeds, such as PlanetLab and Emulab. Remote experimenters can allocate large numbers of nodes in arbitrary combinations, link them with nearly-arbitrary topologies, load arbitrary code for routing, traffic generation and shaping, defense mechanisms, and measurement tools, and execute their experiments. The Emulab software provides sharing of testbed resources among multiple concurrent experiments when enough nodes are available.

Even though the DETER testbed is already capable of enabling researchers to run medium-scale experiments, more nodes are needed, both to enable larger experiments and to handle more simultaneous users. For example, an early DETER experiment on worm propagation dynamics could (just) be squeezed into the then available 72 nodes, but 100 nodes would have simplified the experiment and increased its generality. However, not all researchers are interested in performing very large-scale experiments. One group of researchers used all of the testbed's nodes to perform fine-grain analysis of enterprise networks, complete with actual machines on individual subnets. Having additional testbed nodes available would have enabled them to analyze a large enterprise network.

Given the significant researcher interest in being able to run large-scale experiments, our goal is to build a large-scale

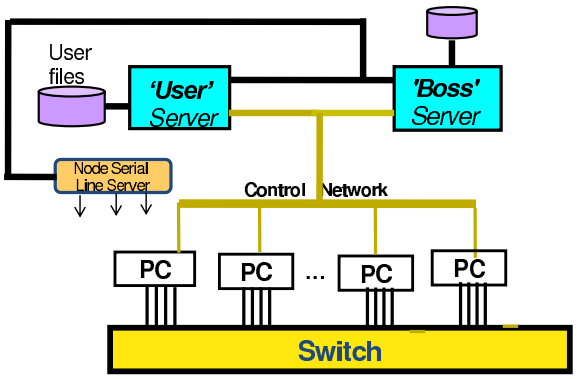

Figure 1: The Emulab Architecture.

testbed facility capable of running these experiments. However, we face significant limitations in available power and cooling resources, and on the maximum floor weight loading in our machine rooms. Thus, our initial solution has been to build the testbed as two tightly coupled clusters running as a single logical administrative domain and interconnected via the CALifornia REsearch Network's High Performance Network (CALREN HPR). One cluster is located at the University of California, Berkeley (UCB) and the other is located at the University of Southern California's International Science Institute (USC/ISI) (See Figure 2).

Using this solution, one potential way of building a larger-scale testbed would be to tightly couple together additional testbeds. However, this approach is a partial solution, as we face the additional limitation that the current instantiation of Emulab on DETER has problems with loading experiments using 200 or more physical nodes.

Instead, we propose an alternate approach as a step toward the *federation* of multiple, independent Emulab testbeds. We define federation as the cross granting of experimenter access and usage rights between separately administered testbed facilities. Effectively, federation enables experimenters to run experiments that span multiple separate testbeds, without the testbeds having to operate under a single administrative domain. Each participating testbed can make its own operational policy decisions and choices and decide whether to admit a new federated experiment or not.

We plan to combine existing Emulab mechanisms for resource reservation and delegation along with extensions that we have developed, to tightly couple simultaneous *experiments* without immediately having to solve the (difficult) issues of tightly coupling *testbeds*. The control information that is passed using these mechanisms, along with the rights they confer and their limitations, are more than mere implementation details. Careful examination of this approach will provide us with useful insights about how to dynamically federate multiple testbeds on an ongoing basis.

The rest of the paper is organized as follows: in Section 2, we provide a background discussion of the Emulab and DETER architectures and firewalled experiments; in Section 3, we present our idea for a prototype federation model and explain the challenges and potential solutions; in Section 4, we discuss our experiences with building the prototype solution; and in Sections 5 and 6, we acknowledge our sponsors and discuss our conclusions.

2 Background

2.1 Emulab Architecture

The basic Emulab architecture consists of a set of experiment nodes, a set of switches that interconnect the nodes, and two control nodes, Boss and Users (see Figure 1). The switches are used to interconnect the experiment nodes. The interconnections are physically separated into a dedicated control network and an experiment network for user-specified topologies. Experiment nodes may be servers, personal computers, sensor motes, routers, or other devices, such as Intrusion Detection Systems, Field Programmable Gate Arrays, etc. Each experiment node has two or more network interfaces, one of which is connected to the dedicated control network. The other interfaces are connected to the experiment network.

The Boss server controls the testbed's operation including the ability to power cycle individual experiment nodes, while researchers log into the Users server to create and manage experiments and to store the data required or generated by their experiments. The testbed's switches are controlled using `snmpit`, a program that provides a high-level object interface to the individual SNMP MIB's of testbed switches. Other programs talk to the power controllers to power cycle nodes, load operating systems onto experiment nodes when requested, and interact with the database to reserve and assign nodes to experiments.

An Emulab experiment consists of a collection of nodes, software to run on the nodes, and an interconnection topology. An experiment is specified using a combination of a `.ns` file and a web interface. The Emulab control software on the Boss server enables multiple, separate experiments to be simultaneously run on the testbed. The software isolates experiments by assigning each experiment to one or more unique Virtual Local Area Networks (VLANs) that connect together the experimental interfaces on each experiment node either using simulated bandwidth-limited and lossy links or using LANs. By using separate VLANs, an experiment's experimental traffic is isolated from other experiments. To prevent one experiment's network traffic from interfering with that of other experiments because of insufficient internal switch or inter-switch bandwidth, the `assign` program is responsible for mapping an experiment's link bandwidth requirements onto the available switch resources in a manner that ensures that the experiment's bandwidth demands match avail-

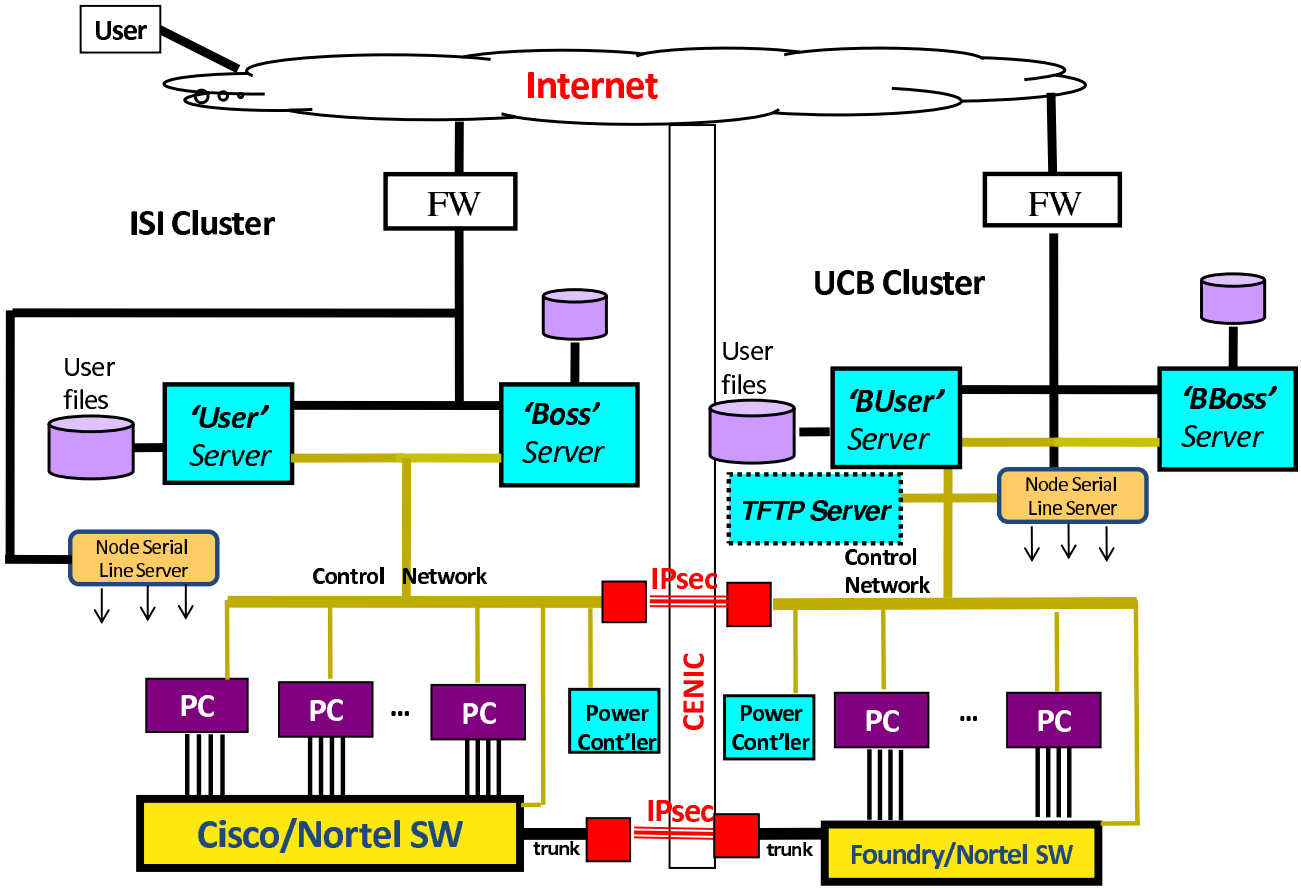

Figure 2: DETER Testbed Architecture.

able inter- and intra-switch bandwidths. Note that unless an experiment is firewalled (as described in Section 2.3), all of the control network interfaces are on the same VLAN.

The Emulab process of swapping in a new experiment consists of several steps: mapping the researcher's desired network topology onto available nodes and switch resources, configuring VLANs on the switches to connect the experiment nodes into the researcher's desired network topology, installing an initial minifs kernel and root filesystem onto the experiment nodes, and then loading and running the desired operating system and software.

In the Emulab trust and privilege hierarchy model, each researcher is a separate *user* of the testbed. Users working together are grouped into *groups*, and a *project* consists of a collection of related groups. Users may also belong to more than one project. Each *testbed* has its own complete (and independent) trust and privilege hierarchy.

2.2 The DETER testbed

To address some of the challenges of building a large testbed, we created the DETER testbed by grafting together two Emulab testbeds (clusters) in a tightly coupled manner (see Figure 2)[1]. The cluster interconnection consists of three network switches connected together by two IPsec tunnels, each carrying entire Ethernet frames including IEEE 803.11q VLAN tags. The interconnection of the two clusters' control "planes" is provided by a virtual wire tunneling Ethernet frames between ports on the two switches Bfoundry1 (at UCB) and Foundry4 (at USC/ISI). The interconnection between the two experimental "planes" is between ports on Bfoundry1 and Cisco4. Each of those switches are connected to other switches, so the large yellow rectangles at the bottom of Figure 2 are not single switches but collections of them.

The two clusters share a common trust structure, with periodic (daily) replication of the Boss and Users filesystems from the USC/ISI cluster to the UCB cluster.

[1] A more detailed description of the DETER testbed can be found in another paper accepted to this workshop.

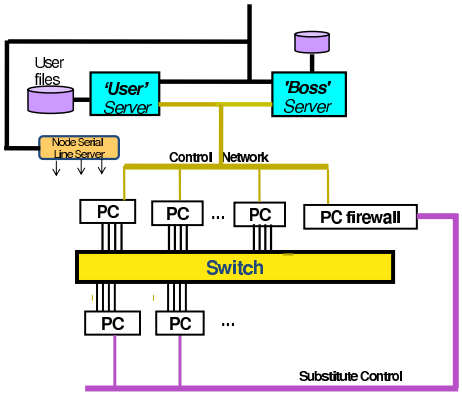

Figure 3: Firewalled Experiment.

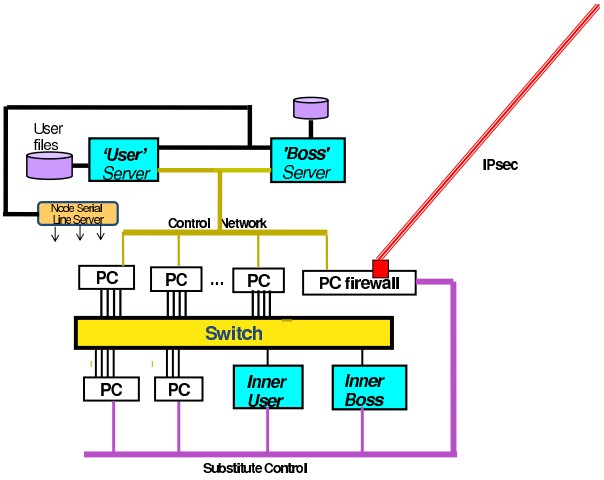

Figure 5: "Half" of a Prototype Federation Experiment.

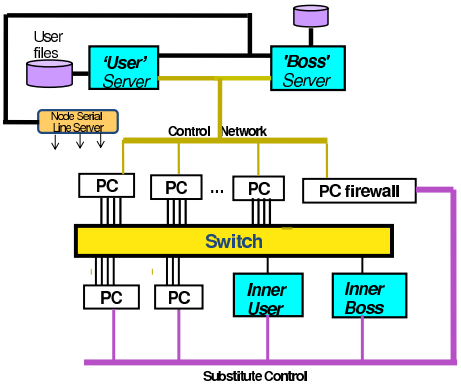

Figure 4: Firewalled Emulab-in-Emulab Experiment.

The two control networks of the clusters use a quasi-static assignment policy for allocating nodes and other resources between them. The serial servers can each only be connected to one of the Boss servers at a time. Both Boss servers are connected to all the switches, but only one of them is responsible for creating and managing VLANs at any given time.

The Emulab process of installing a 3 MByte minifs kernel and root filesystem for a new experiment requires a TFTP transfer and takes approximately six minutes when swapping in across the link between the two clusters. By using a local TFTP server, we are able to reduce this time to two seconds.

2.3 Firewalled Experiments

Because DETER experiments may involve risky code, such as self-propagating worms and virus, experiments must be isolated from external networks, such as the Internet. To provide strong isolation, our approach to enabling federated experiments leverages Emulab's support for firewalled experiments, which enables an experiment to be wrapped up in a boundary-control kind of way (see Figure 3). It is implemented by a smart (layer 2) bridge between the testbed's control VLAN and a newly created control VLAN containing the control network interfaces of the PCs in the experiment. Firewalled experiments are created using an `.ns` file option and `ipfw` rules. It is possible to model enterprise networks with multiple firewalls by creating multiple firewalled experiments on a testbed.

2.4 Firewalled Emulab-in-Emulab Experiments

At a high-level, the Emulab-in-Emulab mechanism lets a researcher reserve a group of experiment nodes and grants them the right to dynamically change the nodes' network topology. More specifically, the mechanism works by making a subset of the Emulab databases and instantiating them on inner Emulab Boss and Users servers created out of two experiment nodes (see Figure 4). The remaining nodes are available for use by the experiment. The researcher is granted administrator rights on the inner Emulab testbed and a login on the inner Boss (*i.e.*, they can become root). The researcher's SSL certificate is used for XML-RPC from the inner Boss to the external (real) Boss, to request VLAN (re)configuration for any node's (experimental) interfaces, and power cycling. The inner Users and Boss servers insulate the external (host) testbed from the trust and privilege structure in the inner testbed, and to a certain extent the exact the version of Emulab running inside (to the extent that one can devise scripts to upgrade/downgrade the schema of the database subset transmitted from the outside testbed to the inner one). As long as the SSH keys are the same, it will still be possible to run experiments, even if we have different users, groups, and projects in the inner and outer testbeds.

From a federation standpoint, an significant advantage of being able to support different versions of Emulab inside and

outside is that it would not be necessary to run the same version of Emulab on different federated testbeds. Finally, the existing firewall mechanisms should provide the same isolation for risky experiments as is currently provided in the DETER testbed when it is connected to the Internet.

3 A Federation Prototype

Our federated experiment prototype is based on the idea of connecting together independent Emulab testbeds by using a modified version of firewalled Emulab-in-Emulab functionality to instantiate subsets of the experiment within each testbed (see Figure 5). This model of operation effectively loosely couples together the testbeds for the purpose of running a large-scale experiment.

In the rest of this section, we first describe how a federated experiment would ideally be performed, and then explore several challenges and potential solutions. We also discuss several hard problems that we have not yet addressed in our federation prototype.

The process of executing a federated experiment proceeds as follows:

1. First, instantiate simultaneous firewalled Emulab-in-Emulab (elab-in-elab) experiments at multiple testbed facilities.
2. Next, co-opt the inner Users and Boss nodes:
   - Designate one set of nodes as the master nodes for the experiment.
   - Ignore User ID assignments and permissions at all nodes, except for the master nodes.
3. The next step is to "implode" the (inner) databases to extract a description of the nodes.
4. Now Emulab's `assign` process can be run on the entire assemblage of nodes. We then separate out all the database state and distribute it to each local site's Boss server, and each local site's Boss server merges everything back in.
5. Then have each inner Boss server request instantiation of the topology at that site.
6. Have each site reports back the assigned VLAN tag numbers.
7. Distribute the necessary disk images from the master to each site's Boss server via `scp` and then have each local Boss server load the operating systems on its local nodes.
8. Then construct IPsec tunnels between the firewalled experiments which translate the tags appropriately. Kevin Lahey at ISI has implemented two independent techniques for doing this; one using the Click router (at our suggestion) and an independent way using the netgraph mechanism in FreeBSD 6.
9. Finally, the experiment runs...

3.1 Challenges and Potential Solutions

Running federated experiments in the wide-area introduces several new challenges, some of which we have already encountered in connecting the USC/ISI and UCB clusters. Here is a description of some of the challenges and potential solutions.

- Running the UDP-based services that Emulab depends on for its operation in the wide area (*e.g.*, DHCP, bootinfo, TFTP, and NFS) might not work and even multicast in the wide area has already been problematic for us. The solution to this problem is easy, each local Boss and Users server provides these services locally. Our idea for DNS is that the `/etc/resolv.conf` file has a search directive listing all the federating experiment suffices, and each local boss has an "NS" reference to the master site for all the other experiment suffixes. For example, it is already the case that boss.elabelab.DETER.emulab.net is a legitimate, resolvable domain name, and thus we could replace "boss" with any other virtual node name in its portion of a federated experiment.
- Collisions in the IP space for unroutable control interfaces could occur. As long as the local (inner) Boss server has sole responsibility for DHCP responses to its nodes and it can reach its outer Boss server, there should be no problems with temporarily renumbering the control net. Note that we have not yet encountered this problem.
- Collisions in the name space of nodes, node_types, OS ID's, and image ID's could occur. One solution would be to append the testbed's domain name to each identifier (*e.g.*, pc3000@isi.deterlab.net), however the length of names might be an issue. An alternative would be to have a table in the database that maps from short prepended identifiers to testbeds. For example,
  - ut<anything> maps to emulab.net
  - wi<anything> maps to wail.wisc.edu
  - cu<anything> maps to cornell.edu
  - vb<anything> maps to vanderbilt.edu
  - isi<anything> maps to isi.deterlab.net
  - ucb<anything> maps to ucb.deterlab.net

- Operating system images for nodes at different sites might not be compatible. This is already an open issue for the existing Emulab testbeds as new types of nodes are added. One potential solution would be to create universal system images that include drivers for a broad set of hardware types. However, differences between nodes may still be an issue (*e.g.* different mappings from the BIOS to COM ports is an issue we previously encountered).

3.2 Hard Problems Not Addressed in an Initial Prototype

While we are confident that we have viable solutions for the problems discussed in the previous section, there are several hard problems that we have not yet addressed in our initial prototype, including synchronizing the swap in of multiple experiments and multiple sites, the requirement for accounts at all sites, and complex permissions and trust management requirements.

The first major problem is the simple requirement to schedule the availability of a major fraction of the available nodes at the participating sites so that they are all simultaneously avaiable. This is, in of itself, quite a challenge given the competition for the resources at key participating sites.

The second problem, swapping in of a single experiment of a thousand nodes among several federated sites in a truly automated way, would require the synchronization of VLAN assigments across *all* the sites. Given the current emulab software, one process must survey all of the vlan tags in use at that moment in all of the switches, and then compute what vlan numbers are available. Futhermore, it cannot allow any other swap-in at any other site to construct any other vlan until all of the vlans are instantiated in every switch. the sites. Obviously, this requirement introduces the potential for deadlock or significant delays if sites are slow in responding or fail.

Our strategy of rewriting the vlan tags allow for each site to contruct its vlans separately and mitigates the problem, but there is still a requirement for synchronization after that is done. The Utah emulab staff has proposed altering the snmpit software so that all vlan assignments would be stored in the database, and the tags computed on the basis of that, at the time of swap-in, which additionally would permit vlan construction to proceed simultaneously in all switches.

The current federation model requires that a researcher have accounts at all the participating sites, however related to the problem of permissions and trust management, the developers at Utah have suggested that permissions and trust management could be pushed down a level. For example, suppose there is a DETER project at Utah's Emulab (www.emulab.net). Then, projects at www.isi.deterlab.net might turn into groups within the DETER project at www.emulab.net. Thus, we could both avoid the problem of requiring accounts at all sites by using a single account, and address the permissions and trust management issue through delegation back to the originating site (and that site's account on the federated testbed). There is still the policy issue of defining which remote testbed's users would be allowed to access a local testbed resources.

4 Our Experiences and Status

In this section we provide an update on our efforts to build a working federation prototype and discuss some of the experiences and lessons learned.

We have implemented support for steps 1 through 3 (see Section 3): site prefixing, inner database implosion, the running of `assign` on the assemblage of nodes, redistribution of consequent database state to the remote sites, identification of cross-campus links and mediation of differing software levels and trust structures between campuses (*e.g.*, running an instance of Emulab-in-Emulab at DETER where the inner testbed software is within a couple of weeks of what is currently running at Utah, and the outer testbed structure is 10 months older than that). Implementing the changes took about 700 lines of changes distributed among a dozen files.

We have succeeded in getting `assign` to process with a single `.ns` file describing nodes on two campuses, and have verified that we have a sufficiently complete list of the tables to be subsetted from the combined database and sent back to each federated site to reflect its share of the experiment (step 4). We have already made the modifications to the swap-in process to enable the rest of the activity that occurs after the assignment process, and tested it with a manual swap in of two halves of an experiment at Berkeley and USC/ISI.

An issue that needs to be addressed in the future is that `assign` uses statically allocated arrays for some characteristics, such as node_type. The limits are unlikely to be reached in federating two or three large sites (*e.g.*, DETER, Utah, and Vanderbilt). A bigger question is the computational complexity of the assignment algorithm and whether it will succeed for 1,000 nodes.

For steps 5 through 6, the Utah Emulab staff has already adopted earlier minor changes we proposed to the VLAN control privileges granted to elab-in-elab experiments so that the inner testbed can request two additional services from the outer Emulab: placing experimental interfaces in trunked mode (something that non-elab-in-elab experiments can already do), and retrieving the list of actual VLAN tags in use at each site so that the tags can be rewritten by the inter-site firewalls. The earlier changes required about 500 lines of new or changed code in 5 files.

The processes of assigning VLANs, loading operating systems (step 7), and replacing the trust structure in the satellite sites, are all working now. We are, at time of the publication of this paper, continuing to resolve some minor details in

the conjoining of the inner control networks (step 8) and the running of the Emulab "events" system.

4.1 A Manual Federation Experiment

While we have made significant progress towards the ultimate goal of automated federation of experiments, we are currently at the state where manual intervention still is required. The necessary technology is in place to allow us to use manual configuration and commands to demonstrate 1,000 experiment nodes interacting on a distributed security simulation.

More specifically, the way it would be done is that at each of the participating sites, we would instantiate separate experiments with separate `.ns` files and then we would manually configure tunneling of the constructed VLANs. The tunneling would require rewriting of the actual VLAN tags using one of the solutions we described above.

If the experiment also requires that the control interfaces in each testbed talk with each other, then it would be necessary to tunnel the control networks together (since the control network addresses are private and unroutable), and it would be prudent to place the each participating group in a firewalled experiment to contain the control network traffic. Placing each federated group in a firewalled elab-in-elab experiment would allow very stringent firewall rules, such as allowing only SSH and XML-RPC traffic from the outside to the federated experiment.

This manual process should be possible to do now, however it would require operators at each site to use the `snmpit` command to place a normally experimental network interface on a node with external Internet access into trunked mode, and then add all the VLANs to be tunneled onto that interface. It would also require punching a hole in the firewall rules to permit UDP traffic between the participating sites.

Kevin Lahey at ISI is currently pursuing a small scale demonstration of this manual approach with the WAIL group at Wisconsin, however he has encountered two problems: they are currently running an older version of Emulab that does not have the necessary emulab-in-emulab features, and the Wisconsin firewall is blocking the UDP ports necessary for tunneling purposes.

5 Acknowledgments

This research was supported by funding from the United States National Science Foundation (NSF), the United States Department of Homeland Security (DHS), and Space and Naval Warfare Systems Center, San Diego, under contract numbers ANI-0335298 (DETER), CNS-0454381 (DECCOR), and N66001-07-C-2001 (DIPLOMAT). Juniper Networks and Hewlett-Packard (HP) donated equipment used by the DETER testbed. Donations were also received from Sun Microsystems and Dell through their University Discount programs.

Opinions, findings, conclusions and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF, DHS, the Space and Naval Warfare Systems Center, San Diego, Juniper Networks, HP, Sun Microsystems, or Dell. Figures and descriptions are provided by the authors and are used with permission.

6 Conclusion

The growing interest in large-scale testing of cybersecurity applications is leading to increasing demand for large testbeds. However, a large testbed requires substantial power and cooling resources from a site and imposes a significant amount of weight loading.

As an alternative to a single large testbed, we have presented techniques for running massive experiments between cooperating Emulab-derived testbed facilities. The experience gained will help us understand the operational and administrative issues with federating testbeds. We discussed the specific steps, several challenges with known solutions, and some open challenges. We also provided a status update on our progress, and outlined a proof-concept experiment that uses manual configuration to demonstrate the feasibility of our approach.

References

[1] BAJCSY, R. *et. al.*. Cyber defense technology networking and evaluation. *Communications of the ACM 47*, 3 (March 2004), 58–61.

[2] BENZEL, T., BRADEN, B., KIM, D., NEUMAN, C., JOSEPH, A. D., SKLOWER, K., OSTRENGA, R., AND SCHWAB, S. Experience with deter: A testbed for security research. In *2nd International IEEE/CreateNet Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities* (March 2006), Barcelona, Spain.

[3] WHITE, B., LEPREAU, J., STOLLER, L., RICCI, R., GURUPRASAD, S., NEWBOLD, M., HIBLER, M., BARB, C., AND JOGLEKAR, A. An integrated experimental environment for distributed systems and networks. In *5th Symposium on Operating System Design and Implementation (OSDI 2002)* (December 2002).