This session discussed the challenges in testbed experimentation and how to overcome them. We compiled the following list:

1. **Experiment realism.** To produce publishable results, experiments must be realistic, i.e., reproduce real Internet conditions. There are three obstacles to realism:
    a. **We don't understand what "realism" means**. This definition depends on the particular experiment, and there is no single "realistic" environment that would fit all researchers' needs. One step towards improving this would be to research common experiment classes and enumerate what aspects of realism are important for them (e.g., scale, legitimate traffic mix, legitimate traffic's responsiveness to congestion, etc.)
    b. **We lack tools to faithfully reproduce components of the real Internet**. For example, there are many traffic generators, but it is difficult to track one that does exactly what a researcher wants. Cataloguing existing tools and measuring how "real" their output is and in what context would help. This is closely related to issue a), understanding what features of a given Internet property (e.g., size of a topology or connectivity) matter for what classes of experiments instead of trying to provide a "realistic topology generator" for all classes. Another issue with existing tools is that they lack a standard interface. Each comes with its own OS/application requirements, set of command line options and settings. It takes time and effort for a researcher to learn how to use a new tool, so many researchers write their own tools or stick to using old tools that they are familiar with.
    c. **We lack data**. A critical condition for replicating realistic environments in testbeds is to have some idea how reality looks! We lack such data in many aspects. There are partial glimpses of Internet topologies but not one generally accepted map. There are a few public traffic traces from small networks, from a very long time ago, or both. They are so heavily anonymized that almost no security or application research can be done with them. There are almost no attack traces for any attacks! We need ways to motivate institutions to share data about traffic, attacks and topology in a safe, anonymous manner.

    Open questions are: do we generate traffic or replay it, how does one model human actions (e.g., human browsing patterns), there is need for benchmarks and metrics for security evaluation; we lack understanding how much large scale matters in different experiments.
2. **Testbed usability.** Researchers would benefit from higher testbed usability. This means development of GUIs, pre-packaged experiment environments (Internet in a box, botnet in a box, etc.) that a researcher can easily load on the testbed and modify; monitoring tools that alert the user when something goes wrong with his/her experiment (e.g., a node fails). Pre-packaged environments are especially interesting since they would significantly improve usage of testbeds, because they reduce barrier to entry. Often, even an experienced testbed user needs a lot of time to generate a new environment for testing whereas in simulations this process is much faster. Further, current testbeds don't offer much with regard to user

support, so a novice user without systems experience needs a lot of learning/warm up time before he/she can really make use of the testbed.

3. **Adversarial testing.** Because security systems in real deployment are challenged by motivated and skilled attackers, evaluating a system's robustness in realistic, adversarial and complex scenarios is of paramount importance. One such adversarial environment can be created through Red team exercises. This is not to replace thorough testing and analysis by authors but should complement it. There is still a lot of resistance in the community towards these exercises. Some downsides are:
    a. **Cost.** Red team exercises are notoriously very expensive. It is difficult to come up with a viable, popular scheme of organizing these in academic environments that guarantees frequency, reduces cost and attracts participants.
    b. **Thoroughness.** Unless very carefully designed, Red team exercises may consist of a collection of random tests instead of systematic testing to verify or refute hypotheses about a system's performance. Some participants in the discussion were wary that if these exercises were a possibility, people would use them to replace thorough testing and analysis, claiming that a solution works if the Red team failed to detect any problems.
    c. **Motivation.** In academic communities, it is difficult to publish papers about testing existing defenses. If Red team exercises are to be done by academics, it is unclear how to motivate participants.

    Some advantages are:

    1. **Better testing**. Because testing is not done by a system's creators, and its goal is to break the system not to show that the system works, it has a potential to uncover new attack vectors and to challenge the system in new ways.
    2. **Possibility of test standardization**. If testing were done by a small number of dedicated teams, they would over time develop tools and environments where novel systems could be quickly integrated. This could lead to evolvement of test standards and security metrics, thus improving overall evaluation of security systems.
4. **Community building**. Current testbeds do an inadequate job of community building. Although there are several large and many small testbeds, a lot of researchers haven't heard of any of them. Many researchers use only one testbed and are reluctant to switch because of the steep learning curve for each new testbed. We need to improve advertising, potentially by organizing regional testbed meetings or even touring institutions and giving talks about various testbeds. An online catalog of testbeds would also help. We need to build testbed communities through workshops (there is one conference on testbeds TRIDENTCOMM, and one workshop CSET), mailing lists, discussion lists, user outreach programs, etc. We also need to motivate collaboration between testbeds. Currently, there is a lot of competitiveness present, and many testbeds operate in isolation. We should foster regular "testbed meetings" where representatives at least from large testbeds (Emulab, Planetlab, DETER, WAIL/Schooner, GENI,

etc) could exchange ideas.

Another issue that closely relates to community building is that it is very difficult to publish papers on metrics and benchmarks, or any aspect of testbed building/maintenance in top venues. These papers just don't fit the focus of many top conferences and journals. This imbalance between on one side the significant work that must be invested in making advances in testbed experimentation and improvement of security evaluation, and on the other side the small chance of publishing this work in top venues discourages testbed research.

5. **Risky experiment support**. Much of security research involves risky actions such as running unknown malware, creating denial of service, breaking into machines, etc. Currently many of these experiments are prohibited or forced to run in isolation but there is large scientific merit in being able to run them in an open environment where interaction with the real Internet is possible. For example, a researcher may want to run the worm within the testbed and collect statistics by having each infected machine report to a specialized server he/she built at his home institution. Today, the server would have to be replicated in the testbed or the researcher would be forced to run a simulated worm, or both. A future direction is to develop flexible methods for risk management that support various open experimentation modes.

   Another issue with open, risky experiments is that they may attract attackers that would retaliate against the testbed. For example, researchers may infiltrate a botnet using testbed machines. The botnet owner could detect this and launch a DDoS attack on the testbed. This creates a lot of problems for hosting institutions. An interesting problem is whether it is possible to somehow hide/separate testbeds from their location so that an attack on a testbed has no ill effects on the institution hosting it.

   Finally, any open experimentation environment requires logging to reduce testbed liability if testbed machines were used for illegal activities. How to provide sufficient, pervasive logging that does not jeopardize user privacy is an open problem.

Future of Testbeds

Mediators:

Sonia Fahmy
Purdue University

Jelena Mirkovic
USC/ISI

Many thanks for input :

Rob Ricci
University of Utah

K Claffy
CAIDA

Paul Barford
University of Wisconsin

Larry Peterson
Princeton University

Wade Trappe
Rutgers University

George Kesidis
Penn State University

Many Research Testbeds

- Emulab, University of Utah
- DETER, USC/ISI and UC Berkeley
- Schooner, University of Wisconsin
- ORBIT, Rutgers University
- PlanetLab, world wide
- GENI, world wide
- ....

Testbed Usage Highlights

	Users	Projs	Exps	Papers	Top Topics
Emulab	2,159	525	12,896	130+	Nw, Dist Sys, Sec
DETER	251	70	2,933	65+	Sec
Schooner	190	20	1,286		Sec, VOIP, Wireless, Prog hw
ORBIT		250	20,000	50+	Wireless
PlanetLab	7,627	670		198+	Dist Sys, Routing, Measurement

Not for comparison, just highlights.
Many numbers don’t have same base (yearly vs total).
Testbed usage depends on type of research done on them.

Testbed Usage Highlights

- Emulab helped set up 19 other testbeds internationally
- DETER provides unique tools for user-friendly, security experimentation and for federation with other testbeds
- Schooner provides unique environment for botnet experimentation
- ORBIT usage ~90% daily, LRU allocation system
- PlanetLab carries 2-4 TB of traffic daily towards 1M outside addresses

Not for comparison, just highlights.
Testbed usage depends on type of research done on them.

Testbed Usage Highlights

	Papers	Public Testbed	Private Testbed	Sim	Could use public testbed
SIG 07	35	7	12	7	6 no change 2 traffic realism 2 scale 2 topo realism 5 hw
Sec 07	23	0	3	2	1 no change 1 hw 1 risky
NSDI 07	27	11	8	3	8 no change 1 hw 1 scale

Open Problems

- Environment realism
  - Realistic (real?) background and attack traffic
  - Large-scale support or scale-down techniques
- User-friendliness and outreach
  - Tools, user support, packaged experiments
- Testing realism
  - Red team testing
- Community building
  - Releasing packaged experiments for a paper
  - Valuing papers on testbeds, experimentation, validation of others' experiments
- Risky experiment support

Summary

- Environment realism
    - Realistic (real?) background and attack traffic
    - Generators vs replay
    - How do we model people actions?
    - We need benchmarks, metrics, better anonymization, data sharing
    - Large-scale support or scale-down techniques (federation)
- User-friendliness and outreach
    - Tools, user support, packaged experiments, advertising
- Testing realism
    - Red team testing, collaborative testing and collaborative research
- Community building
    - Releasing packaged experiments for a paper
    - Valuing papers on testbeds, experimentation, validation of others' experiments
- Risky experiment support
    - Currently isolated, we need to support controlled, open experiments
    - Liability implications large, can we hide ownership of testbeds?
    - Need accountability and logging in testbeds