Testing the Effectiveness of Large-scale Cyber Defenses[1]

J. Just[2], G. Kesidis[3], K. Levitt[4], Jeff Rowe, Felix Wu, Phil Porras[5]

Abstract

The need for good large-scale cyber-defenses is enormous. It arises in many areas: from protecting priority flow customers on residential broadband networks [3] *from service theft to protecting critical infrastructure from cyber-terrorist attacks. There are numerous research projects and, even, some products that are concerned with defenses against large scale attacks as worms and DDoS. The nagging question is how effective are these defenses? In principle, they can be deployed on operational networks to determine how they perform on the next worm, for example – but only if there are guarantees that their deployment is safe. Besides the issue of safety, deployment of these systems is expensive as a defense system for large-scale attacks must be at many locations to be effective. If such deployment determines the defense is effective, it is only for the single (or few) attack that was observed. Thus there is a clear need to evaluate such defenses before deployment. There is a surprising lack of science that can be applied to testing cyber-defenses designed for large-scale deployments, e.g., enterprise or Internet scale. This paper is submitted to provoke discussion on the pressing need to evaluate such systems. It presents a framework for evaluation that includes analytical reasoning, simulation, emulation and actual operational deployment.*

1. Introduction

Good large-scale cyber-defense systems are desperately needed. To satisfy this need, there is considerable effort underway to develop systems that can defend against worms, DDoS attacks, and attacks on the Internet infrastructure. What unifies these systems is both their need for thorough testing in an operationally meaningful environment, e.g., the live Internet, and the impossibility of such testing. Networks, in general, and the Internet, in particular, exhibit chaotic behavior beyond a certain scale so size and scale do matter. No experimenter can launch the variety of attacks that are likely to be seen in the wild over time. While cyber-defenses, assuming they are safe (not obviously the case), can be deployed on a trial basis, their evaluation will only be against attacks that are successfully launched during that trial period. So it is important to examine issues of how the performance of the defense varies (1) with both the scale of the attack and the deployment of the defensive system and (2) with attacks that are outside the range of tested or observed attacks.

Both simulation and emulation approaches have been and are being used to test network and some host defense related research, e.g., Emulab [10] and PlanetLab [11]. In an effort to

[1] This work was partially funded by the projects indicated below The views and conclusions contained in this document are solely those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any department of agency of the U.S. Government.

[2] J. Just (jjust@globalinfotek.com) is with Global InfoTek, Inc. and is funded by Navy SPAWAR contract Contract N66001-04-C-8007.

[3] G. Kesidis (kesidis@engr.psu.edu) is with the EE and CSE Depts of Penn State and is funded by the NSF/DHS EMIST project NSF 0335241.

[4] K. Levitt, J. Rowe and F. Wu ({levitt,rowe,wu}@cs.ucdavis.edu) are with UC Davis CS Dept. and are funded by the NSF/DHS EMIST project NSF 0335241

[5] P. Porras (porras@sdl.sri.com) is at SRI International and is funded by the NSF/DHS EMIST project NSF 0335241

achieve more thorough testing, several government programs are developing cyber-testbeds (such as the DHS/NSF DETER/EMIST project [6], [7], [8]). There are enormous legislative and regulatory pressures to deploy almost any such system. What, if any, set of cyber-defense technologies should be deployed? Can any affordable testing help solve the problem?

Given the practical impossibility of testing defenses on a realistically-sized network with enough time to adequately explore an infinite attack space, any affordable testing must be small scale relative to the Internet or even large enterprises. Hence, a key question, if not the key question, is, "Given the results from small-scale testing of any specific cyber-defense technologies, should anyone be willing to commit the investment funds needed to deploy such defenses on a large scale?" Test results reflect specific network and host configurations, attacks, test procedures, etc., Can we scale the test results to enterprise or Internet size networks with any confidence? Can we generalize the test results to other types of attacks? If so, how is such scaling or generalization to be performed? Can any cyber-defense technologies be compared meaningfully in such tests?

These are the important testing questions that confront all researchers in the cyber-security arena. Unfortunately, they raise far too many issues to be answered as part of any single research project or program. To be fair, these are very difficult questions and the state of the field does not make them easy to deal with. Availability of test data, e.g., "realistic" generated host and network traffic, is problematic. Organizations do not collect, let alone disseminate, real world data that could be used to develop and test theories for many reasons such as privacy concerns, security through obscurity and avoiding possible embarassment.

Consequently, as the literature shows, researchers usually ignore them and publish their small scale test results. They never answer the "So what?" question. Roy Maxion has discussed this problem from the perspective of characterizing intrusion detection systems [5].

The paper discusses why these questions are increasingly important, suggests what the goals of such testing should be, describes key aspects of the problem that make it so complex, and offers some suggestions on how to proceed. The discussion focuses on worms and worm defenses, but the questions raised apply to other large scale attacks.

2. Goals for Cyber Defense Testing

Benchmarks are sets of well specified tests that are publicly available, vendor neutral, and generally accepted by the community for comparing the certain performance aspects of competing products in a particular problem domain. Benchmarks capture some, but not all, key aspects of that problem domain and the results of benchmark testing are verifiable by third-parties. We believe that cyber defense testing will benefit greatly from a series of benchmark tests for different aspects of the problem. Unfortunately we are nowhere near being able to perform acceptable custom testing, let alone benchmark tests.

Meaningful testing of cyber-defenses implies results that are indicative of what would happen in the real world if there were significant attacks on systems protected by those defenses compared to those that were not protected. The numbers and types of tests required to obtain meaningful results for large scale deployment of cyber defenses is enormous. Again the goals of such testing should be determining how the performance of such defenses scale with the size of attack and magnitude of deployment (e.g., are there emergent properties) as well as how well the defenses perform against previously unseen, unknown attacks. We are not after magic or crystal ball visions but rather scientifically validated theories that provide some ability to predict results.

Given the lack of theory, it is important to have, at least, tests that are appropriate to the hypothesis under test. For example, there are major differences in requirements for testing worm

defenses vs. denial-of-service attack defenses vs. defenses against BGP, DNS or other network protocol attacks. Reproducible test results are equally important. At a minimum, each test of an attack-defense pair must have a well defined network topology and configuration (including hardware, software, and policy configurations on all hosts and network components), appropriate background traffic generation and attack traffic generation, specific defense mechanism configurations, and well defined metrics appropriate to the attack-defense scenario.

Min developing theories, models representing different system elements will play a key role in both the scale-up and scale-down arguments and in robustness to classes of attacks rather than just tested attacks. From simply a scientific testing perspective, we believe the following steps are likely to be necessary:

- Unit level testing of components to generate individual engineering models of the performance of sensors, estimators, controllers and response mechanisms given different inputs and host/network environments.
- System level testing of integrated components on small networks (sub-LANs) to generate engineering models of the system level performance of sensors, estimators, controllers and response mechanisms given different inputs and host/network environments. These results should be compared to what would be predicted from the individual component models from above.
- System level testing of integrated components on LANs to generate simplified more aggregated engineering models of the system level performance of sensors, estimators, controllers and response mechanisms given different inputs and host/network environments. These results should be compared to what would be predicted from the small scale system models from above.
- Since it is unlikely that there will be universal deployment of any defense system, it is important to have validated models of how unprotected hosts and networks respond to attacks and how these unprotected hosts and networks interact with, or subvert, the protected hosts and networks.
- Testing beyond the LAN level is likely to involve emulation (modeling, virtualization) of many key aspects of the system and the network being simulated. These results should be compared to what would be predicted from the small scale system models from above and should also be compared, where possible, to results of similar attacks observed in the wild. Models should be updated to reflect and capture
- Since testbeds are usually limited to several hundred to several thousand network nodes (components including end-systems), tests beyond tens of thousands of nodes begin to involve much more simulation of behavior than actual generation of such behaviors. Validation of models against real world events becomes more and more important at this level.

3. Impediments

3.1. Lack of Science

In A.S. Tanenbaum's book *Computer Networks* (3rd Ed., Prentice Hall, 1996, p. 555,556), he says "Unfortunately, understanding network performance is more of an art than a science. There is little underlying theory that is actually of any use in practice. The best we can do is give rules of thumb gained from hard experience and present examples taken from the real

world." This statement is not universally accepted as fact but it certainly represents the current state of knowledge well.

Large scale testing of cyber defense technologies represent the some of the most ambitious testing ever undertaken by any organization because of the enormous size of the test space, the inherently discrete and non-linear nature of the defense systems under test and their computer network environment, and the lack of coherent underlying theories that span deterministic and probabilistic behavior. Is it necessary to build a vast knowledge base of experimentation to support the development of such theories or are there reasonable approaches that can be taken with small number of tests coupled with emulation, simulation or analytic results.

3.2. Characteristics of the problem and questions

1. Behavior of software, hosts and network ranges from deterministic to probabilistic
2. Hosts and networks are complex discrete systems that do not have the continuity properties of linear time invariant systems.
    a. Deterministic system transform functions are very useful for characterizing system and require much less testing to prove results but most hosts and networks are not linear time invariant systems.
    b. What theories can guide us in determining when we can use deterministic testing and when to use statistical testing for cyber defense technologies involving hosts and networks?
3. Can we apply a law of large numbers to some aspects of complex discrete systems?
4. This is not just a statistical design of experiment problem. We lack underlying theories and bodies of knowledge to guide us. How do we build up that body of knowledge? What specifically is missing?

4. What is Needed

1. *Realistic Data*: A common approach to testing cyber-defense technology is to analyze or replay traces of captured data. Network traffic captured from the Internet, for example, provides a source of realistic background data for a particular site. Running this captured data through passive intrusion monitors can provide a valid test of an algorithm's performance with respect to false alarms. Testing the ability to detect attacks in the presence of background in this manner, however, is more problematic. For repeatability, a single background data trace must be combined with a variety of attack traces. Simple time ordered interleaving of events is insufficient due to the interference effects that many attacks will have on normal traffic. What is needed is a model of attacks, a model of background data and a model of the interference between the two. An even more difficult case comes when evaluating active defense technologies; where the background data must be modified to include not only interference from attacks, but effects reflecting the actions of defense systems themselves.
2. *Characterization of Networks*: As has been pointed out by Floyd and Paxson [9] providing connectivity between heterogeneous networks is one of the main reasons behind the success of the Internet. As a result, validation of the performance claims of a cyber-defense system in one network cannot be generalized to other sites. What is needed is a method for classifying salient features of network topologies, and a way

to judge whether features of the test network are relevant to a specific environment in which the system will be deployed. A further difficulty comes from the time evolution of Internet traffic in general; traffic behavior changes drastically between day and night, between weekday and weekend and continuously and irreversibly as new network applications are deployed.

3. *Inability to Launch Real Attacks*: For obvious reasons, testing on the live Internet must be done without using malicious attack code. While some types of mild attacks, such as scanning, might be tolerated, no sane person would suggest testing malicious Internet worms against cyber-defenses deployed on the Internet. What is needed is a way to test system connected to live networks, where real background is present, with attack simulators. These simulators would produce traffic with the relevant features present in the real attack considered, in a way that doesn't exploit any system vulnerabilities.

5. Illustrating the Problem

The previous section highlighted the requirements from a strictly testing perspective. This section explores the size of the attack and defense space.

5.1. Illustrative Attack Space Complexity

We will use worms to illustrate the complexity of the attack space even though they represent a small (but potent) portion of the potential cyber-attack space. Table 1 summarizes possible worm target selection or propagation strategies. Other key worm characteristics include the vulnerability exploited and the worm payload. Table 2 highlights the different strategies that an attacker might employ using worms as the basic attack mechanisms. The possibilities for non-worm attacks are much greater.

Table 1 Basic Worm Types

Target Selection Strategies of Worms	Description (Example)
Random Scanning	Repetitively selects a "random" IP address and tries to infect them (Slammer, Code Red I). Variants include different address generation schemes such as PRNG, permutation scanning, binary search
Hit List	Repetitively selects from pre-compiled list of known IP addresses for vulnerable, accessible hosts and attacks them.
Topological	Uses information on local infected host to identify next targets for attack (e.g., email addresses, URLs, buddy lists, rhost lists). Variants include metaserver worms that obtain addresses for next attack from community locator services (e.g., directories of hosted game servers or search engines) and contagion worms that spread infection among vulnerable application nodes (e.g. Internet games, instant messenger, media servers & players) by riding normal traffic to/from servers or peers
Hybrid	Combinations of the above (e.g., Nimda)

Table 2: Worm Attack Characteristics

Characteristic	Examples
Targeted population and vulnerability	Specific version application or OS, client or server, type of vulnerability exploited, size and demographics of target population

Table 2: Worm Attack Characteristics	
Characteristic	**Examples**
Spreading mechanism (including propagation vector)	Email, instant messaging, web traffic, ftp, direct UDP or TCP (no vector), etc – each has different topological implications
Targeting(propagation) algorithm	Random scanning (single or multi-stage, local/regional/ global), topological (from local address books, buddy lists, rhost tables, etc.), contagion, targeted lists (precompiled hit lists, permutation scanning w/wo partitioning), Warhol (optimized hybrid of hit-list and permutation scanning), etc
Speed of propagation	Very fast, slow (contagion-like), variable or staged
Communications and control	Fire and forget (none), phone home (w/wo encryption), self-organizing networks (based on visit history), pre-arranged "message center", updating (load new worm type, hit lists, new payloads, new instructions, etc)
Size and signature	Static or constant, polymorphic, metamorphic, programmable and updatable
Modes and staging	Spectrum from single atomic worms to complex, multi-mode, multi-vector, multi-stage worms
Payloads	Spectrum from benign to corrupting process to installing Trojans or spyware to reformatting hard disks to subtle corruption of key data to reflashing of bios to ??

5.2. Illustrative Cyber Defense Technologies

The literature contains many examples of defensive technologies and components that can be assembled into cyber-defense systems. Table 3 summarizes a few of them.

Table 3 Illustrative Cyber Defense Components		
Sub-System	**Function**	**Example**
Sense (monitor)	Behavior of application (e.g., number of threads, resource utilization, new process forks)	HACQIT application monitor
	Content of app. I/O (e.g., access to files, memory locations, registry, processes, network)	Balzer Wrappers
	Behavior of OS (e.g., abnormal pattern of system calls,	Forrester System Call Monitor (Stide)
	Integrity of files, registry, file structure	Tripwire
	Log/audit entries for monitored actions	Emerald BSM
	Performance of host (e.g., QoS, statistical resource anomaly or logical anomaly)	Orincon
	Behavior of processes, ports	
	Performance of network (e.g., latency, congestion, unusual traffic patterns on ports)	
	Content of net traffic (e.g., unusual traffic, excessive ICMP traffic, known bad content)	Snort
	Anomalies in BGP	Dartmouth sensors
Communicate, Control, Analyze	Out of band communication channels for monitoring and control	
	Sensor alert correlators	
	Sandbox testing/analysis	HACQIT learning
	Cross-enclave correlators (	
	Enclave controllers	Alpha-LADS
	Enterprise network controllers	
	Response analyzers	
Respond	Application level I/O content filters	

Table 3 Illustrative Cyber Defense Components		
Sub-System	**Function**	**Example**
	Wrappers for prevention/virtualization	
	Host firewalls	Embedded firewall (ADF)
	Enclave/site firewalls	
	Routers	
	High speed packet inspection/modification	CloudShield
Recover	Switch to uncompromised real or virtual host	
	Restore file system integrity	
	Rejuvenate compromised application	HACQIT
	Start replica of service and state (also wide area)	ITUA
	Wide area failover	

5.3. Testing Requirements

There are two key requirements for delivering defensible test and evaluation results in any project (i.e., defensible both to the teams participating in the tests and to outside reviewers). One is that high fidelity attack behavior and false alarm behavior must be delivered to all of the sensors, correlators, analyzers, containment mechanisms, and recovery processes used by the technology development teams. Note that any testbed is small compared to the Internet. The second requirement is that the large-scale attack behavior and the performance of the protection system on the test network must be at least representative of what would be seen on real enterprise networks and the Internet with similar attack types. Validating test procedures involves ensuring that both of these requirements are met.

5.4. Criteria for Testing Large Scale Cyber Defenses

Metrics need to be appropriate to the problem but should include aspects of attack resistance or containment, false alarm rates, defense overhead and other impacts, and recovery times from successful attacks.

6. Conclusions

These are testing questions that confront all researchers in the cyber-security arena. Unfortunately, they are far too large to be answered as part of any single research project or program. Consequently, researchers usually ignore them and just publish their small scale test results. They never answer the “So what?” question. The paper discusses why these questions are increasingly important and what aspects of the problem make it so complex statistically and analytically.

As noted in *The Code of Best Practices—Experimentation* [1]: “As noted earlier, enterprise operations are too complex and the process of change is too expensive for organizations to rely on any single experiment to “prove” that a particular innovation should be adopted or will provide a well-understood set of benefits. Indeed, in academic settings no single experiment is relied on to create new knowledge. Instead, scholars expect that experimentation results will be repeatable and will fit into a pattern of knowledge on related topics. Replication is important because it demonstrates that the experimentation results are not the product of some particular circumstances (selection of subjects, choice of the experimentation situation or scenario, bias of the researchers, etc.). Placing the results in the context of other research and

experimentation provides reasons to believe the results are valid and also help to provide linkage to the causal mechanisms at work.

"An experimentation campaign is a series of related activities that explore and mature knowledge about a concept of interest. They use the different types of experiments in a logical way to move from an idea or concept to some demonstrated operational capability. Hence, experimentation campaigns are ways of testing innovations in an organized way that allows refinement and supports increased understanding over time."

We believe that it is very important to put large scale testing on a much more scientific basis and that more than one research program should be started to focus on the problem and begin building the scientific basis for testing with scalable results.

7. References

[1] Alberts, David S., "Code of Best Practice for Experimentation", DoD Command and Control Research Program, July 2002. Retrieved 27 March 2003

[2] Haines, Joshua, et al. "Validation of Sensor Alert Correlators", IEEE Security & Privacy, Jan-Feb, 2003, Vol 1, No 1.

[3] Hamadeh, Y. Jin, S. Walia, G. Kesidis and C. Kirjner, "Pricing and security for residential broadband access," in Proc. CISS, Princeton, March 2004.

[4] Lippmann, R.P. and J. Haines, "Analysis and Results of the 1999 DARPA Off-Line Intrusion Detection Evaluation," Proc. Recent Advances in Intrusion Detection (3rd Int'l Workshop RAID 2000), H. Debar, L. Me, and S.F. Wu, eds., Springer-Verlag, New York, 2000, pp. 162–182;

[5] Maxion, Roy A. and Kymie M.C. Tan. "Benchmarking Anomaly-Based Detection Systems", 1st International Conference on Dependable Systems & Networks: New York, New York, USA. 25-28 June 2000.

[6] Benzel, T., et.al., "Cyber Defense Technology Networking", Communications of the ACM, March 2004.

[7] Web site for DETER project: http://deter.isi.edu

[8] Web site for EMIST project: http://emist.ist.psu.edu

[9] Floyd, S. and V. Paxson, "Difficulties in Simulating the Internet", IEEE/ACM Transactions on Networking, 2001. 9(4): p. 392-403

[10] Emulab, http://www.emulab.net/

[11] PlanetLab, http://www.planet-lab.org/