SANDIA REPORT

SAND2008-7122
Unlimited Release
Printed October 2008

Using Emulation and Simulation to Understand the Large-scale Behavior of the Internet

Helgi Adalsteinsson, Rob Armstrong, Ken Chiang, Ann Gentile, Levi Lloyd, Ron Minnich, Don Rudish, Keith Vanderveen, and Jamie Van Randwyk

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2008-7122
Unlimited Release
Printed October 2008

Using Emulation and Simulation to Understand the Large-scale Behavior of the Internet

Helgi Adalsteinsson, Rob Armstrong, Ron Minnich, Don Rudish, Keith Vanderveen
Scalable Computing Research and Development Department
Sandia National Laboratories
P.O. Box 969, Mail Stop 9158
Livermore, CA 94551-9159
{hadalst, rob, rminnich, dwrudis, kbvande}@sandia.gov

Ken Chiang, Levi Lloyd, Jamie Van Randwyk
Computer and Network Security Department
Sandia National Laboratories
P.O. Box 969, Mail Stop 9011
Livermore, CA 94551-9159
{kchiang, llloyd, jvanran}@sandia.gov

Ann Gentile
Visualization and Scientific Computing Department
Sandia National Laboratories
P.O. Box 969, Mail Stop 9152
Livermore, CA 94551-9159
gentile@sandia.gov

Abstract

We report on the work done in the late-start LDRD *Using Emulation and Simulation to Understand the Large-Scale Behavior of the Internet*. We describe the creation of a research platform that emulates many thousands of machines to be used for the study of large-scale internet behavior. We describe a proof-of-concept simple attack we performed in this environment. We describe the successful capture of a Storm bot and, from the study of the bot and further literature search, establish large-scale aspects we seek to understand via emulation of Storm on our research platform in possible follow-on work. Finally, we discuss possible future work.

Contents

1 Intro

This is a summary report of the work completed under the latestart LDRD *Using Emulation and Simulation to Understand the Large-Scale Behavior of the Internet.* This project ran from July 2008 through September 2008.

Due to the short duration of this project, this work was intended to further elucidate the problem space, potentials for application and analysis, and programmatic possibilities.

2 Project Definition and Long-term Scope

The short-term goal of the LDRD was to develop a research platform for studying the dynamics of the Internet. In the long-term this platform can be used to emulate both normal activities and attack scenarios, with the specific intent of enabling capabilities for detecting and predicting attacks and for understanding attack evolution. If successful, it is possible that this platform could be used to study even non-attack evolutionary behavior of the internet.

Scale is key to the building of such a system from which truly representative behavior will emerge. Thus, we intend to build an internet simulation on a scale that approaches the size of a small country. Our goal is to enable realistic cyber scenario testing and further develop our understanding of the results with an Internet Emulator of the same nation-state scale. The Emulator will be used to study cyber security scientifically with both theory and verifying experiment together. Previously experimental data for Internet-under-attack needed to be captured "in the wild." With this Internet Emulator every aspect of the attack can be held under scrutiny, helping to validate models of the attack at scale. Such a facility might also suggest strategies against attacks and aid in reconstructing network topologies and conditions hidden from observation.

Understanding the Internet at scale is critical. Like the power grid and the economy, the Internet has grown into a complex ecosystem, and just like other complex systems, behavior at the largest scales cannot be directly inferred from the characteristics of the smallest, individual computers. Examining the constituents of the system is not sufficient to understand the emergent behavior at large scales.

Initial focus for this research will be on the dynamics of simple cyber-pathogens. The long-term goal is to model and emulate the Internet on the scale of Estonia with the aim of understanding the propagation and mechanisms of network-born viruses and attacks. This required us to develop the means to emulate multiple types of computers, from servers to clients to workstations to laptops, connecting and disconnecting in different places. Understanding the impact of individual computer configuration and network topology will likely suggest mitigations and defenses that are not apparent at the level of the individual computer.

An Internet-scale Emulator will be built from a state-of-the-art cluster and virtual machine technology to verify and aid the construction of simulations. We have developed an Internet emulation based on virtual machine technology and high-performance computers. We have created a system that will support 10's to 100's of desktop-like operating systems per node and, coupled with a state of the art supercomputer, will permit 10^5-10^6 networked virtual machines. This amounts to a realistic emulation of the Internet for a country the size of Estonia. The Emulator will inform and verify the simulation. While simulation and emulation of only simple cyber-pathogens will be considered for the brief period of this work, the focus will be on proving the potential of this approach to make a case for follow-on funding.

Mathematical characterizations of the system will be defined that may eventually enable run-time attack detection. The large-scale nature of the problem lends itself to employing statistical methods for characterization of the systems behavior, which, in turn can be used to provide probabilisitic descriptions of the system's state. Even without a detailed understanding of the attack scenarios, the resultant effect of an attack on the system may manifest itself as a low probability state or as a state change. We will detect and correlate abnormalities and state changes in the emulate system with controlled attack scenarios in order to learn detectable attack signatures.

This work will eventually give us a "bird's eye" view on the Internet. Our current situation is one of standing in the middle of a forest, observing a flame, and trying to determine if we are standing near a campfire or in the midst of a forest fire: we have no idea of the global state of the network. Indeed, gaining such a global view is impossible, since information about the state of the network is transmitted over that same network; frequently, cyber security attacks include code designed to obfuscate what is happening. It is simply not possible to perform a controlled experiment of the Internet, on the Internet. With the technology we propose, we can emulate and simulate the current nation-scale Internet, and craft effective cyber security techniques based on a global view, not a merely local view.

Due to time limitations, in this late-start proposal, we will build a proof of concept system that demonstrates the fundamental required elements for the ultimate goal: 1) scalability of the system and 2) the ability to induce dynamic behavior representative of attack scenarios. Once we have established this basic system, we will propose future work to extend the system to involve more sophisticated emulations of internet entities, their interactions, and attack scenarios.

3 Internet Emulator

The key component of this work was the development of a research platform to be used for understanding the Internet and attacks at scale. Using Virtual (VM) technology on a large, high performance compute cluster, we can emulate millions of networked machines. Each virtual machine can run the actual (not reduced or simulated) system environment. This enables one to study the actual protocols and the software stack targeted in the attack.

Our Emulator provides previously unavailable potential for studying not only large-scale attacks but also the efficacy and consequences of possible responses. Via the virtual machines, we would have full access to the system state data, enabling a system-wide understanding. This would allow the application of statistical methods for analysis, which would be otherwise ineffective with only limited or incomplete data such as would be obtained from only access to a few machines "in the wild". Further, we can test potential counter-measures in a realistic environment, which could reduce the potential for unintended consequences that may occur as a result of applying a response based on a small-scale or reduced study to the actual Internet.

The Emulator VM's are specially crafted to have a low memory footprint, that was adapted from the Lguest[1] VM, in order to support the running of hundreds of VM's per node.

In our proof-of-concept work, our Emulator ran on 50 nodes, with 100 Virtual Machines per node to result in a total of 5000 emulated machines. We conceptually divided the machines into two separate networks, each of 2500 machines. All VMs ran standard Linux, *not* a simulated kernel. We wrote and ran a trivial worm, that stats on one machine at boot and "attacks", via a open socket, other machines at random. The "attacks" were designed to have 100 percent success rate, provided that the attacked machine is not already infected. Further, the mechanics of the selection of hosts to attack were such that the attack could spread from machine to machine within a network with far greater ease than from one network to another. We have made a small GNUplot based animation from the attack. The purpose of this proof-of-concept work was not to emulate a truly sophisticated worm, but rather to develop and prove the technology necessary for supporting N-to-one VMs on a single node and to obtain system state data.

As a result of this study, we are ready to attempt to run more sophisticated attacks on the Emulator, with the caveat that most of the common Internet attacks target Windows, not Linux, and therefore there are access and licensing issues to be worked out to enable running 5000 Windows instances.

4 Investigation of Actual Large-Scale Attacks

In order to determine characteristics particular to large scale attacks, we researched particular malware samples captured in the wild to understand their communication mechanisms.

Samples of the Storm Trojan[2, 3, 4] were captured using open source malware capture software known as Nepenthes[5]. The Nepenthes software works by emulating vulnerabilities on a network, as malware try to exploit these vulnerabilities, they are captured. The Storm bot (which is the rootkit deployed by the Storm Trojan after infecting a host) was then ”cultured” by running it on an isolated testbed of vulnerable computers.

We observed the Storm Trojan on our isolated testbed to understand its command and control protocol. Storm uses robust, encrypted P2P communications to achieve resiliency against failing central command and control nodes. Because of its P2P communications architecture, Storm is difficult to observe using an isolated environment. Much of our analysis of this bot has been supplemented by white papers in the open community.

By analyzing and understanding the communication mechanisms of malware species like Storm in depth, it becomes possible to replicate these communication mechanisms on a larger scale, using Internet Emulation, to observe/model/analyze the threat posed.

5 Large-scale Aspects for Study

Early attacks were independent one-on-one affairs. While that model still remains effective, more modern attack architectures, such as botnets, consist of a large number of entities. The level of interaction of the entities is variable, however the possibilities of so many entities, with access to so many compute resources, and so much bandwidth, under a central control are great.

While there are a number of interesting issues for potential study in the mechanics of threats in general, this work specifically targets increasing the understanding of aspects either key to or unique to the large-scale nature of the attacks. Thus, for instance, mechanisms by which entities in a large-scale botnet, such as Storm, hide from anti-virus software are *not* aspects that are a focus of our study, except for cases where such mechanisms may result in signatures that are discoverable by statistical analysis enabled by the large number of entities in the system. In this section we discuss aspects of large-scale, possibly coordinated, attacks that we have elucidated both from the capture and cultivation of a Storm bot and from literature studies that we will seek to study in the emulated environment in future work.

We divide these aspects into two classes: non-behavioral aspects and emergent behaviors. The distinction is that non-behavioral aspects consist of mechanisms or architectural issues, whereas behaviors constitute manifestations of system dynamics.

5.1 Large-scale Non-Behavioral Aspects

Due to the changing nature of the machines in the system, at any given time, new machines may be infected or already infected machines may become uninfected or disconnected. Therefore botnets must consistently updating their knowledge of their constituents. Mechanisms exist within Storm for a new node to notify the botnet of its existence and for unresponsive nodes to be dropped from the botnet. This information is innately stored within a distributed hash table. Such non-behavioral mechanisms which are necessary to ensure the functioning of a large scale system would be a target of our further study.

In order to remain resistant to offensive counter-measures that exploit a flaw in the code base and to increase their own obscurity, the attack code base is often altered, either externally, or through even simplistic, though effective, random code changes (e.g., location of jumps). Such mechanisms would be particularly necessary in hiding a large-scale attack where statistical examination of a large number of infections could reveal similarities that could be used to discover the attack. We thus intend to investigate further adaptive code and methods for discovery. It is anticipated that simple code changes may not be sufficiently obscuring and would be still discoverable by statistical methods. For instance, statistical data analysis methods have been used to trace the origin of spam even when random word changing techniques have been applied.

5.2 Large-scale Emergent Behavior

We are particularly interested in large-scale emergent behavior. This is behavior that requires a significant number of entities to be present in order to manifest itself. Emergent behavior can be of two types. The first is planned behavior, where the behavior is explicitly written into the system and the emergence comes from the required interaction of multiple elements of the system in order for the behavior to take place. An example of such behavior in Storm is the coordinated Distributed Denial of Service (DDoS) attack that may be instigated by a reaction to attempts to deliberately multiply download bots (as one would do in an attempt to capture a bot for study). Note, that this is not invoked as an 'intelligence' requiring response, but rather is invoked upon threshold-crossing counts of probes. For the DDoS to be truly successful, large numbers of bots must act in concert. In this case however, the mechanism of the attack and the interaction (which is minor) of the bots are explicitly planned.

Of perhaps more interest is emergent behavior that is not planned, but comes about as a side-effect of the of the multiple system entities. While a detailed example of this for Storm is currently unknown to us, since we have not yet emulated the system at large-scale, one can envision possibilities based on some of the known mechanics of Storm. For example, bots choose a pseudo-random id upon joining the system. When a significant number of entities have joined the system, it is inevitable that there will be id collisions. In such a case, attempts to hone in on an particular entity by id, may result in misdirection to the identically numbered entity. This problem is exploited in counter-measures against the botnet by hash poisoning, in which a known legitimate bot id is duplicated by a non-bot site in order to redirect or absorb traffic intended for the original bot site. Hash poisoning aside, one can for see that such misdirection could result in unintended side-effects of the system.

As stated earlier, the level of sophistication of coordination in attacks is small. The distributed hash table mechanism, of course, has a reasonable degree of coordination, but even in the DDoS, there is not coordination in the actual details of the attack, but only in that the bots are attacking the same site. This lack of sophistication is not a flaw, as Storm is very effective in what it seeks to do, and more sophistication in the bots would result in a larger footprint that would perhaps make them more easily discovered or harder to insert into a system. Nevertheless, the potential for more sophisticated coordination lurks in the future, possibly through multiple entities with agent-like capabilities that could reside on machines owned by the attacker, that could monitor the system state and issue complimentary commands to various parts of the botnet in order to achieve an overall effect.

6 Future Work

In this section we highlight some areas of possible future work.

6.1 Internet Emulator

Now that we have identified some large-scale behavioral and non-behavioral aspects of interest in Storm, we would like to run Storm on the Emulator. We believe that the Emulator not only provides previously unavailable possibilities for studying truly large systems, but also that the Emulator's essential structure allows for studying aspects that could not be studied in other ways. For instance, Storm's own anti-detection defensive mechanisms involve alterations of the Windows OS that we believe can be sidestepped by interaction and control possibilities of the virtual machine technology.

Also, we wish to explore performance, CPU contention, and bandwidth contention issues from co-existing a large number of virtual machines on a single compute node.

6.2 Large Scale Aspects of Malware

We would like to do further studies in behavioral and non-behavioral aspects in malware, including but not limited to Storm. Additionally, there is an intriguing belief that the Storm botnet is being subdivided and sold [4]. It would be interesting to study (1) if the resulting sub-sections "evolve" differently and (2) how the large-scale capabilities are dependent on the size and configuration of the botnet e.g., does the subdividing of Storm change any of its possible behavioral aspects.

6.3 Data Analysis

A key feature of the large-scale Emulator is the potential to obtain total state data of the entire system. This gives us the potential to apply statistical analysis techniques to the study of this data. Collecting, processing, and representing large volumes of frequently changing data to enable such analysis is itself a difficult task. In future work, we will utilize the capabilities of Sandia's OVIS[6] project which has been successfully used for real-time monitoring and analysis of live, frequent data streams of high performance computational clusters. The OVIS tool is another capability that uniquely positions Sandia to address this problem space.

References

[1] *Lguest: The Simple x86 Hypervisor*, http://lguest.ozlabs.org

[2] P. Porras, H. Saidi, and V Yegneswaran, *A Multi-perspective Analysis of the Storm (Peacomm) Worm*, CSL Technical Note, (2007).

[3] C. Kanich, K. Levchenko, B. Enright, G. Voelker, and S. Savage, *The Heisenbot Uncertainty Problem: Challenges in Separating Bots from Chaff*, First USENIX Workshop an Large-Scale Exploits and Emergent Threats (LEET '08), April 2008.

[4] see http://en/wikipedia.org/wiki/Storm_botnet and references therein

[5] http://nepenthes.mwcollect.org

[6] *The OVIS Tool for Scalable, Real-time Statistical Monitoring of Large Computational Clusters*, http://ovis.ca.sandia.gov

DISTRIBUTION:

1	MS 0899	Technical Library, 8944 (electronic)

Sandia National Laboratories