Understanding the Behavior of Internet Worm through PArallel Worm Simulator (PAWS)

Tiffany Tachibana
Computer Science and lnformation Technology
California State University, Monteray Bay
ttachibana@csumb.edu

Graduate Mentor: **Jia Bai** and **Xiaowei Li**
Faculty Mentor: **Prof. Yuan Xue**

Originally Submitted on: July 31, 2010
Updated on: February 21, 2013

TRUST Research Experiences for Undergraduates (TRUST-REU)
in Cyber Security and Trustworthy Systems 2010

Institute for Software Integrated Systems
Vanderbilt University

Understanding the Behavior of Internet Worm through PArallel Worm Simulator (PAWS)

Tiffany Tachibana

Abstract

Internet and the computer networks have become crucial to efficient operation of institutions and the society. But, this makes it possible for a lone hacker to significantly impact these institutions and society through worm attacks. Such worm attacks are becoming common news leading to a feeling of insecurity. The intention of this research is to study the worm's propagation behavior. These worm spread studies are adapted from the work of several computer security experts and researchers. Contained and isolated yet a realistic environment is required to study the propagation behavior of worms. DETER and PAWS are simulation tools used in this research experiment to study the Slammer worm's propagation.

1) Introduction

The ubiquitous growth of the Internet has been successful in changing many institutions and social interactions. The availability of high-speed local area network connections and other type of network access advancements in the world have significantly increase the adoption of an Internet life-style. Due to the realization that Internet activities are very convenient, fast, informative and cost-effective way of communication, the majority of the institutions have structured their business online. Thus, highly valuable information passes through the network pipeline. These transfers of data entice the attackers to steal or destroy valuable information and disrupt the normal network connection for fun, "fame" or an amount of cash. One of the attacker's tools to infiltrate such stealthy attack is through computer worms.

A computer worm is a self-replicating program that propagates itself across the network. This self-replicating program uses an algorithm to find its target. Once it discovers the target, the worm sends out a probe to exploit the security flaw of that target machine and transfer a copy of itself. The worm's copy will activate a new cycle of propagation. By inspecting the history of computer worm attacks, since Morris worm rose in 1988 [1], Internet worms have been a vast threat in network security. Its fast spreading characteristic and considerable amount of damage is alarming. For example, the Slammer worm infected 90 percent of vulnerable hosts in ten minutes. It was the quickest worm outbreak in history. Slammer worm unleashed a single 404- byte user datagram packet (UDP) to port 1434 that runs Microsoft's SQL Server and Microsoft Server Desktop Engine (MSDE) last January 25, 2003. Exploiting this vulnerability, the worm

infected 75,000 hosts and caused network outages due to high-volume scans and bandwidth traffic. Even though Slammer worm has no malicious payload, it caused significant disruption to financial, transportation, and government institutions [3].

The objective of this research is to understand how certain Internet worms behave. A better understanding of worm's behavior can help to predict future worm's traffic, speed and threat. To obtain this, simulation alternatives are needed to study the propagation of Internet worms. In this paper, simulation alternatives such as mathematical model, testbed and realistic internet-wide experiment are described and were based from the contributions of well-respected computer security researchers [1][5]. To understand how worms spread, computer security researcher like Zou et al. studied the worm modeling propagation using mathematical techniques to study biological diseases [5]. DETER [6] is a testbed used for real machine testing that can run a self-replicating program in an isolated and contained environment. For a realistic internet-wide simulation of worm spreads, PAWS is an excellent tool used in the experiment to replicate Slammer worm's propagation.

2) How worms spread?

Many computer security researchers [1][2][5] used epidemiology models to analyze how worms spread. Zou et al. stated that computer viruses and worms are analogous in their self-replicating and propagation behaviors. Thus, the mathematical techniques developed for the study of biological infectious diseases can be adapted to the study of computer viruses and worms propagation [1]. The simple epidemic model assumes that each host stays in one of the two states either susceptible or infectious. Susceptible hosts are hosts that are vulnerable to infection while infectious hosts are host that have been infected and can infect others. In this model, the number of contacts between infectious hosts and susceptible hosts is proportional to the product of S(t) = the number of susceptible hosts at time and I(t) = the number of infectious hosts at time [1].

3) DETER Testbed

The cyber-Defense Technology Experimental Research (DETER) testbed provides publicly available infrastructure for conducting computer security experiments such as denial-of- service, worms, viruses and other malicious program. DETER testbed is designed for isolation to prevent the experiments from colliding with each other. It also assures containment to prevent exfiltration of packets from the testbed [7]. DETER is easy and very convenient to use due to the following aspects:

3.1 Web-based interface:

Allow users to fill a form to begin an experiment and upload an *ns* script file or use the user-friendly JAVA GUI to create a network topology. The JAVA GUI version allows the

experimenter to configure the nodes with its corresponding operating system and the LAN's bandwidth size, link loss and others.

3.2 Remotely Access:

Each interconnected nodes created in the experiment are real machine that can be configured and re-imaged through Secure Shell (SSH) connection.

4) PAWS: The Internet-scale worm spread simulator

Songjie Wei and Dr. Jelena Mircovic developed a single, configurable, customizable, realistic and versatile Internet-wide simulator tool called PAWS [5] for worm propagation studies. It is designed for a high-fidelity simulation of a worm that can faithfully simulate each worm replication's behavior and interaction with the Internet environment. Described below is PAWS's simulation design:

4.1 The Internet Model:

The worm spread simulator tool replicates a realistic Internet topology the Internet at the Autonomous System (AS) level. PAWS obtains its global connectivity data from Route Views project [4]. The Route Views data supplies the actual copy of Border Gateway Protocol (BGP) routing table of participating AS-routers. The information of the routing table can be used to derive the peering relationship between the inter-AS connectivity that satisfies the need of Internet level fidelity.

In order to achieve realistic scanning results, PAWS designs the Internet routing by distributing the routing information, where each node stores only the routing table of AS it simulates. When a worm scan is generated, the simulation nodes determine the destination ASes. The nodes look on its routing table to calculate the shortest path to the destination AS and bandwidth consumption on their path [6].

4.2 The Worm Model:

PAW is designed to simulate different worm spread events in common PCs. It is customizable and configurable, depending on what particular Internet worm propagation an experimenter wishes to study. The PAWS's program builds a worm description file that defines the worm's features and network environment where users can specify the worm's: (1) scanning rate, in scans per second, (2) scanning strategy (e.g., random, uniform or subnet), (3) worm's transport protocol (TCP or UDP), (4) vulnerable population size and distribution (uniform or lognormal), (5) size of a worm scan, in bytes, (6) infection delay, and (7) lifecycle of worm [5].

4.3 The Worm's Distributed and Discrete Simulation Model:

Each node establishes a stream socket and connects to all other simulation nodes [6]. The simulation tasks are shared by multiple physical machines, which exchange their local results over a network. Each node simulates a portion of the whole Internet and divides the processing and communication loads which balances the CPU cost among the nodes. Thus, PAWS's authors described this as distributed simulation [5].

PAWS worm-scan behavior is a time-discrete procedure. Every time unit, each infected host produces a list of target Internet Protocol (IP) addresses to scan. PAWS accumulates the worm scans in one destination machine. The simulation nodes' cross-traffic uses stream sockets to exchange data at the end of each simulation interval (one second). The scanned non-vulnerable hosts or non-routable IP addresses are dropped or processed at the sender-side machine, thus the inter-node communication is lessened [5].

5) PAWS Experiment

The DETER testbed and PAWS worm spread simulator, were the two simulation alternatives that are sufficient enough to implement a real world experiment in studying Internet worm propagation. In this experiment, PAWS program is configured to simulate the worm spread of Slammer worm. Table 1 shows the worm's feature and the values are based from the Slammer worm modeling study of D. Moore et al. and Zou et al. [3][1].

To setup the experiment on DETER testbed, five machines namely node0, node1, node2, node3 and node4 are connected to a switch (LAN). The attacker is set to the master node0. The client machines are set to nodes 1 through 4, also known as simulation nodes. Using SSH, *paws.tgz* file was extracted at node0. Two executable files are run on node0 as well. The paws server file starts the master node and *slammer.sh* will start the four slave or client nodes.

Default Internet Model	IPv4 address space
Scanning_rate	4000 per second
Vulnerable_ population	75,000
Function Determine_vulnerable_host	Randomly mark 10 vulnerable hosts as infected
Simulation Interval	0.05 per second
Scanning strategy	random
Transport Protocol	UDP
Life cycle	10 minutes

Table 1. Slammer worm's features

5.1 Vulnerable Hosts Scan Result

After running the executable commands, there will be an inter-node communication during the initialization process between each simulation node. During this stage, the function determine vulnerable host() is executed to create vulnerable hosts by scanning the IPv4 address space which is 4,294,967,296 IPs. The distribution of vulnerable hosts on each node is roughly the same. Fig.1 shows the total vulnerable hosts scanned at node3. Totally, there are 80,143 vulnerable IPs.

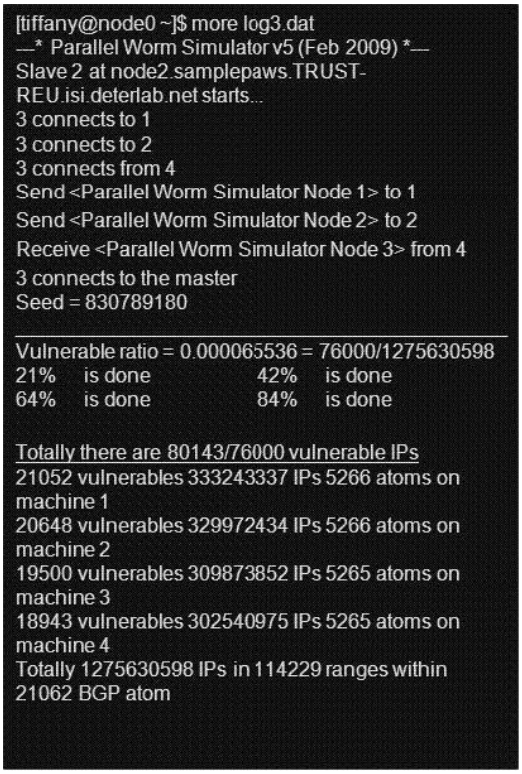

Figure 1. Total number of Vulnerable Hosts

5.2 Infected Hosts Scan Result

After the worm scan, log files are created for each simulation nodes. Fig. 2 displays the number of infected hosts in every second I(t) graph of the Slammer worm scan results. The y- axis displays the number of infected host and the x-axis indicates the time in seconds. The total infected hostsof the four simulation nodes is 79,376 in 600 seconds among the 80,143 vulnerable hosts.

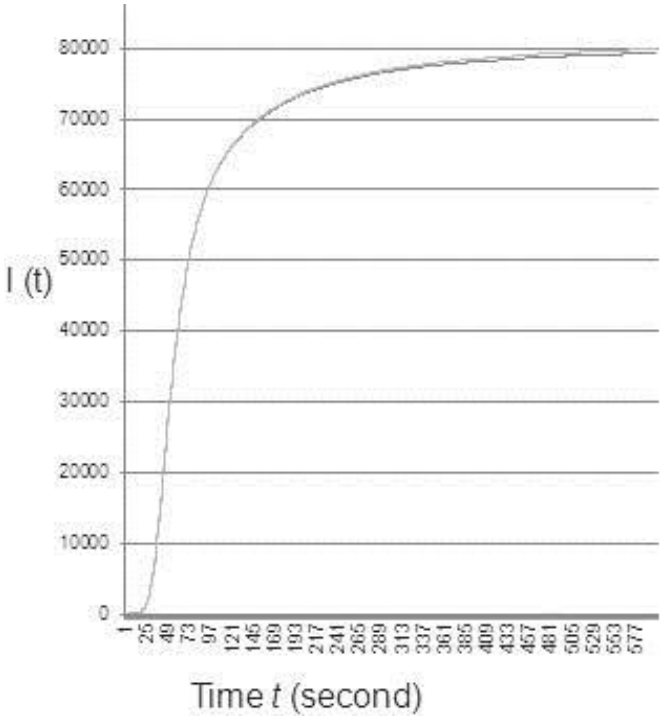

Figure 2. Infected host at time t I(t)

5.3 Observations

At early stage, the worm's growth incredibly multiply and infected almost 66,000 vulnerable hosts or IPs in two minutes. Slammer worm's propagation undoubtedly quick due to its characteristics. It randomly scans the target IPv4 address space and drops non-vulnerable IPs. It uses UDP as its transport protocol to send the copy of worm, thus a non-connection-oriented protocol does not bother to resend the dropped packets. Also this experiment is homogeneous. Thus, no patching strategy is involved. However, the number of infected hosts slowly increases at the latter stage of the worm propagation. As the number of infected hosts grows, the number of worm scans increases and creates scan traffic.

Therefore, the Slammer worm created network congestion, with limited bandwith, causing the connected hosts to crash that led to denial-of-service attack.

6) Conclusions

This worm modeling experiment was meant for students who took Network Security class with worm modeling and dynamic quarantine studies. Without such background, learning Internet worm modeling methodology is a challenge. Through this research experiment, the following observations can be made: (1) Studying the trend and traffic of worm propagation will give the defender a sense of strategies if such worm's traffic can be blocked or slowed through slow human intervention. (2) PAWS is capable for a realistic Internet-scale worm spread

simulation that can mimic different types of Internet worms, due to its configurable and customizable feature. (3) Further research focuses to on how other Internet worms propagate, how software can be developed to generate potential worms and how to develop effective countermeasures is necessary.

Acknowledgements

My sincere gratitude goes to: (1) Prof. Mooi Choo Chuah for her patience and guidance in helping me perform the experiment, (2) TRUST-REU program for this excellent research opportunity, (3) Vanderbilt University research mentors, namely Prof. Y. Xue, J. Bai, X. Li, and (4) Dr. Sathya Narayanan at CSUMB for all the help.

References

[1] W. G. C. C. Zou and D. Towsley. Worm propagation modeling and analysis under dynamic quarantine defense. In ACM CCS Workshop on Rapid Malcode.

[2] G. M. V. D. Moore, C. Shannon and S. Savage. Internet quarantine: Requirements for containing self-propagating code.

[3] S. S. C. S. S. S. D. Moore, V. Paxson and N. Weaver. Inside the slammer worm. IEEE Security and Privacy, 1(4):3339, July/August 2003.

[4] U. of Oregon. Route views project. http://www.routeviews.org/.

[5] J. M. A. H. S. Wei, C Ko. Tools for worm experimentation on the deter testbed. http://www.isi.edu/mirkovic/publications/trident09.pdf.

[6] M. S. S. Wei, J. Mirkovic. Distributed worm simulation with a realistic internet model.

[7] D.T. Benzel. Deployment, and use of the deter testbed. http://www.isi.edu/deter/docs/200708-usecdw-deter-design-deploy.pdf.