Discrete Modeling Simulation of Worm Propagation and Comparing Worm Infection Using Random Scanning and Local Preference Scanning

Avinash Khetri (2K18/IT/035) | Shreoshi Roy (2K18/SE/120)
Delhi Technological University, Delhi, India
Department of Information Technology | Department of Computer Science
avinashkhetri_2k18it035@dtu.ac.in | shreoshiroy_2k18se120@dtu.ac.in

Abstract

One of the most malicious malware infecting computer systems by depleting system resources, tampering with important data, and installing new malware in today's time, are worms. They are one of the fastest propagating malware because of their self sustaining nature and can cause rampant damage to computer networks within days. Our project addresses the need for simulating worm propagation in medium-scale networks so that it is easier to understand how exactly a worm infects computers and what are some of the factors affecting its spread. By taking inspiration from [1], we have developed a light-weight, user-friendly tool that simulates how a worm infects various computers in a medium-scale network. We have used the discrete-event simulation model to build this tool and have also compared the results of worm propagation through random scanning and local preference scanning.

Table of Contents

1. INTRODUCTION

A malware, shorthand for malicious software, is a piece of computer code that is aimed at extracting personal benefits at the expense of a host's computer. This can include damaging or modifying data, locking important files, stealing sensitive information, or overtaking host privileges to install any software. Although malware poses an acute threat to every computer that exists on the internet, the only type of malware most people are aware of is viruses. But actually, worms are more dangerous than viruses because unlike viruses, they do not need any host and can start replicating without any interaction with the user. Hence we have developed this tool so people can get a better understanding of worms and subsequently take more measures to protect their computers from worm attacks.

1.1 What are worms?

As defined by [2], worms are a type of malware that is independent and does not require any host or software to attach to. They replicate very fast and can infect all the computers present in a network in a very short period of time. They take advantage of a computer or network's vulnerabilities to creep into a system and cause damage to it. They target computers that have certain shortcomings such as systems that are not updated, systems with a lack of security protocols, etc.

Even though when worms were created by a Cornell student, they were harmless and were only created to point out the security loopholes in an operating system, the worms in today's time have become very advanced and inconspicuous. They can cause harm to its victims in multiple ways as stated in [3], like seizing control of the system from the user, installing and launching unsolicited programs from the internet, triggering Distributed Denial Of Service attacks, and encrypting files of the system to get a ransom. Such incidents of worm infection have incurred billions of dollars worth of losses for companies and individuals in the last ten years and continue to spread rampantly.

1.2 How do worms spread?

Once a worm enters a system, it will act as a host and the worm will scan all computers on its radar to find more potential victims. It can find all the users in a computer's network or users with whom the host has interacted in the past to connect with them and look for vulnerabilities automatically. Once it finds another suitable system the worm will copy itself into the new system and can now use that system as a new host. This way a worm follows recursion to propagate into a large number of computers in exponential time [4].

Worms usually target whole networks instead of attacking a single computer so that they can create huge botnets [5]. They may spread by sending random emails or messages with suspicious links that can activate the worm if a user clicks on it. They can also impersonate media files and crawl into systems via file sharing. While the earlier versions of worms followed a random order of propagation, the newer ones follow more advanced techniques to replicate themselves even faster. Thus in our study, we simulate both random scanning as well as local preference scanning to compare their results.

1.3 Motivation

Malware analysis can be done in a variety of ways like analyzing the file (comparing Hash Codes, checking strings, etc), running the file in a virtual and restricted environment, or performing a mixture of both. These processes may be tedious and time-consuming and may not even be the best tools for teaching or research as some actions or network calls may not occur in certain conditions or students may not be adept enough to explore these methods accurately. Hence, teaching complex concepts of malware and generating the interest of students is always a challenge. But research has shown that visualization and simulation have proven to be the best techniques for educational purposes and are the best ways for the comprehension of new ideas [6].

Hence we have created this simulation-based study which is aimed at improving the understanding of worm propagation and explaining the main idea instead of getting into the cumbersome details and procedure required to actually infect a computer with a worm and then studying its spread. This tool is very easy to install, beginner-friendly, and can be used by every student to grasp the concept of worm propagation.

1.4 Objective

The objective of this paper is to create a tool that shows a simulation of how worms connected via a network propagate when a single computer of the network gets compromised. We have used the discrete time-event mathematical model for simulation [7].

For the sake of simplicity, 100,000 (Ω) computers have been assumed to be connected out of which 1000 (N) of them are considered to be susceptible to worm invasions. These susceptible computers have been considered to have successive IP addresses in groups of ten. That is, the computers prone to worm attacks have IP addresses between 1 to 10, 1001 to 1010, 2001 to 2010, and so on. Hence, for every 1000 computers, there exists a group of 10 computers with successive IP addresses that can be attacked by worms successfully and there are in total 100 such groups.

Initially, only one computer is assumed to be infected, say having an IP address of 1001. After an interval of one time unit, it starts looking for other computers in the network that can be attacked. The rate at which a computer can scan other computers is called scan rate(η). If a vulnerable system is found in the scan, then that system will become infected and can act as a new host in the next timestamp. This process keeps occurring recursively until all the 1000 vulnerable computers get infected.

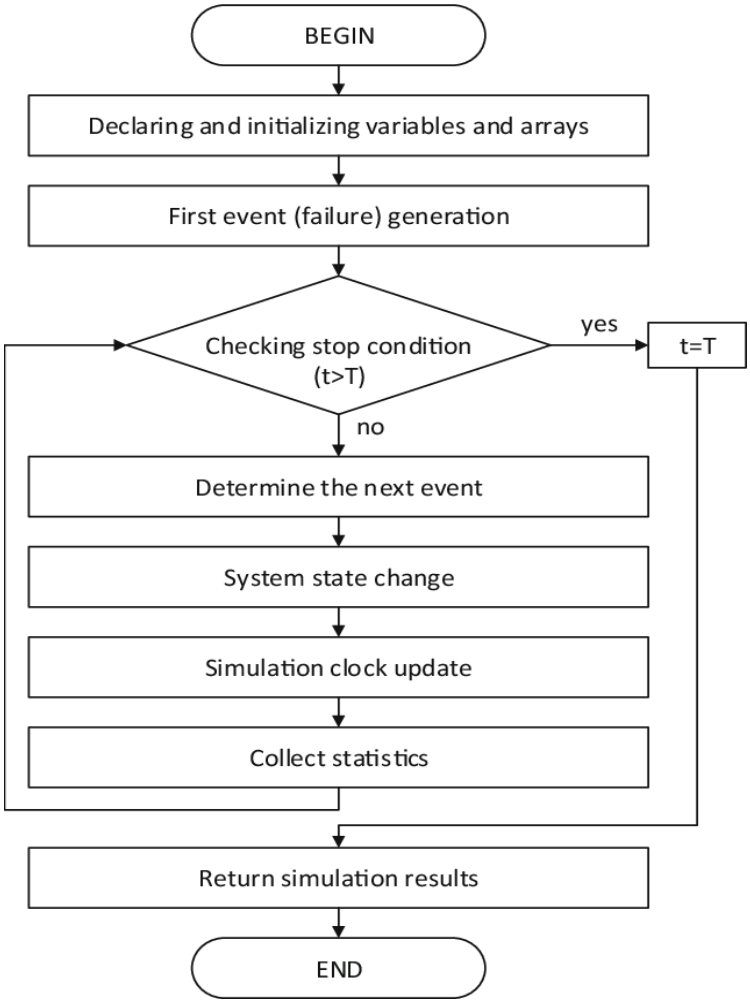
Further, we illustrate the comparison of simulation results obtained when a worm propagates through random scanning vs local preference scanning through graphs.

2. METHODS

This section describes the various models and theories used for this paper and can be categorized into four subsections namely- Discrete-event simulation model, Worm propagation mechanism, Random Scanning method, and Local preference Scanning method. All concepts and definitions for these methods have been explained below.

2.1 Discrete-event simulation model

A discrete-event simulation model as defined by [8] is a model that considers time to be discrete instead of continuous. Time is sliced into equal-sized intervals and the state of every variable can change only when the next time interval occurs. Any change in the state of a variable is called an **event** and no event can occur between two consecutive timestamps [9]. Hence the system can jump to various states or events instead of occurring in a continuous fashion. That is why this model is also called next-event time progression.



Reference- [10] https://www.researchgate.net/figure/Flowchart-of-the-discrete-event-simulation-model_fig1_325197422

2.2 Worm propagation mechanism

To understand the worm propagation mechanism, we have derived heavily from the **Epidemic Model**, an extension on the SI (Susceptible Infected) Model which was devised to study the transmission of viruses and communicable diseases in humans which turn into epidemics or pandemics. This is a differential equation model consisting of homogenous networks, that is, all systems are considered to be identical. They are assumed to have the same design and all systems are equal. Any computer can directly communicate with the other to propagate an infection. The network of computers is visualized as a completely connected graph [11].

According to this model, the worm propagation at any given time can be represented by the equation:

$$\frac{dI(t)}{dt} = \frac{\eta}{\Omega} I(t)[N - I(t)]$$

Where $I(t)$ = Infected systems at time t

Ω = Total number of systems in the network

N = Total susceptible systems

η = scan rate

2.3 Random Scanning

Random scanning is a method used by worms to look for potential victims in the host's network. All computers are considered to have an equal probability of getting infected. The host selects a computer at random, which can be vulnerable as well as non-vulnerable. The random IP address generated could also be non-existent that is no host exists with that IP address [12]. Thus Random Scanning is a hit or miss technique. This method is the most frequently used method for scanning when the Epidemic model is being considered. Some of the famous worms which spread through random scanning are Slammer and Code Red.

2.4 Local Preference Scanning

Although random scanning is a simple technique for worm propagation, it is not the most ideal. The more advanced versions of worms use Local Preference Scanning as it is faster. In this method, there are a number of sub-groups or sub-networks within the network. As discussed in [13] and [14], the computers having IP address closer to that of the infected host fall in the same sub-group and have a higher probability of getting scanned as compared to other computers. The infected host searches for more vulnerable systems within this sub-network. This method is more effective as susceptible computers are not always equally distributed into the IP space and sometimes various security measures may prevent a scan to reach distant IP addresses. This scan is even more effective if the sub-groups formed are larger in size. Some of the famous worms which spread through local preference scanning are Sasser worm and Blaster worm.

3. DESIGN

The entire Codebase has been written in python 3 with libraries such as matplotlib, numpy and Tkinter. The GUI has been implemented using Tkinter library which helps run the Python code in a GUI tool so it is compatible with most operating systems. The data has been visualized in the form of graphs by using the Matplotlib library.

3.1 Implementation Details

A sample space consisting of Ω=100,000 and N = 1,000 computers has been considered to exist in an isolated network. Vulnerable systems are those systems are those which do not have updated patches installed so they are susceptible to worm attacks.

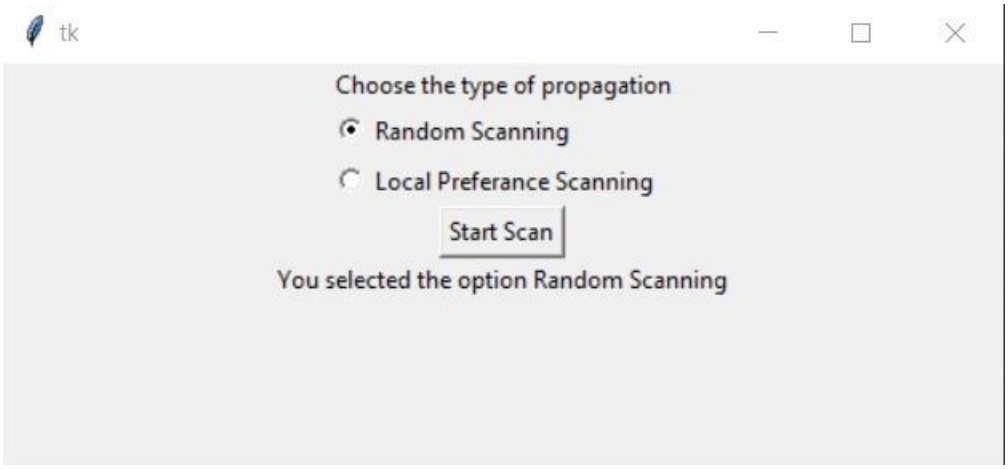
Further assumption is that the worm starts to propagate infection within the network initially from 1 machine which has IP address of 1001.The scan rate(η) of infected machine is 3.This implies that a worm-infected computer can scan to 3 other IP addresses in the network at each time step. A vulnerable system is infected immediately if a worm finds it and this newly infected system can also, in turn start spreading the worm from next step. In this way the worm propagates and infects the whole computer in a network.

We need to find the number of infected computers at each time step t(t=1,2,3...) which is represented as I(t). We simulate the worm propagation 3 times to get the three vector of the number of infected IP, I(t). The simulation ends when all the vulnerable machines are infected. At initial point, I(0) = 1. We have implemented simulation of two kinds of scanning:

For Random Scanning, an infected computer x randomly selects another 3 IP addresses with in the entire IP address space in a time unit. For Local Preference Scanning, an infected computer at each time step with IP value x picks the target IP address value y by the following rule:

1. With probability p = 0.8, it picks a random value y such that $y \in [x-10, x+10]$

2. With probability p = 0.2, a random value y between 1 to 100,000 is picked



3.3 Limitations

The simulation is not precise and exhaustive. Many worms follow strategies other than random and local preference scanning. Hence, not all situations can be evaluated using simulation. Also, this tool creates only a simulation and does not provide any counter-measures that a user can take to prevent or sabotage a work attack.

Real life networks can vary in size and structure. The given simulation provides insight to a normal network with 100,000 IP address space. Networks in real life might give slightly varying answers depending upon arrangement and size and severity of the worm. The variables can be tweaked till they match the given conditions. However the general trend would follow for the result.

4. RESULT

This section illustrates the results we derived from the simulation of both random scanning as well as local preference scanning methods. These results are intended to demonstrate how the method of scanning can affect the worm propagation rates of a general worm and not intended for an exhaustive or definitive study of propagation by a particular worm. They may differ slightly in each simulation run and may also vary depending on the specific type of worm under study.

We conducted three simulation runs for both of our methods and have visualized the data in the form of a graph as seen in figure 1 and 2. We can observe that both the methods result in an ‘S-

shaped curve' or a Sigmoid curve with only one inflection point. Initially, the spread is slow as only a few vulnerable computers have got infected but the propagation increases exponentially as more and more computers get infiltrated and start acting as hosts. They keep on increasing until all the vulnerable systems in the network have been infected after which the graph shows a straight line parallel to X-axis at 1000 systems.

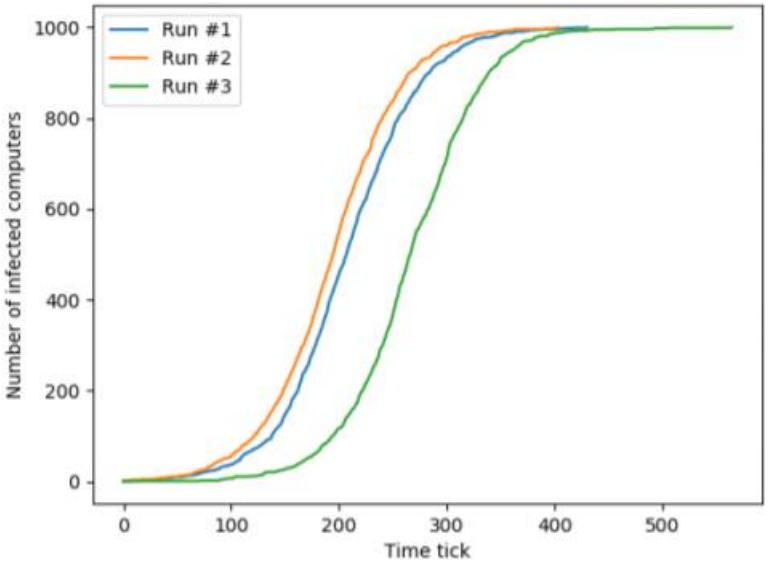


Figure 1: Three simulation runs for worm propagation through random scanning method

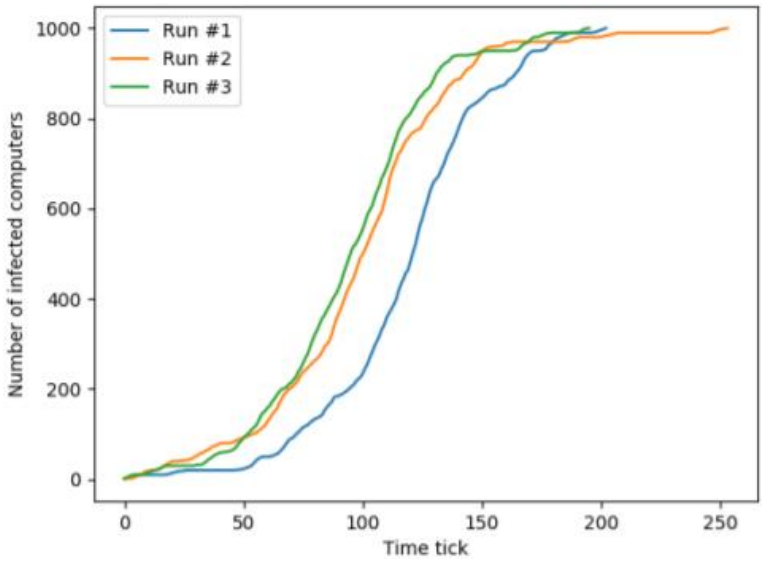


Figure 2: Three simulation runs for worm propagation through local preference scanning method

We also recorded the number of time intervals required by a worm using random scanning and local preference scanning to propagate into all susceptible computers and have tabulated the

results in Table 1. During our three simulation runs, random scanning took 431, 404, and 565 time intervals respectively to infect all vulnerable systems whereas local preference scanning only took 169, 267, and 183 time intervals respectively for the same. Hence, as expected local preference scanning is a clear winner when it comes to spreading worms faster in a network, especially when vulnerable systems are not uniformly spread across the entire IP range. Even the graph of local preference scanning shows how the worm propagation takes off at an earlier stage in all the simulation runs as compared to a more uniform spread of the random scanning method.

Simulation Method	**Time intervals taken to spread to all vulnerable computers**		
	Run 1	**Run 2**	**Run 3**
Random Scan	431	404	565
Local Preference Scan	169	267	183

Table 1: Time intervals taken to spread to all vulnerable computers

5. CONCLUSIONS AND FUTURE WORK

We have designed and developed a light-weight and user-friendly tool which can be easily installed and run on any operating system to simulate the propagation of a worm in a medium-scale network of computers. We have successfully implemented the discrete-event model and the Epidemic model according to our use case of worm propagation. Through this simulation, we have demonstrated how a worm propagating using local preference scanning can spread faster than a worm propagating through random scanning. The results show a qualitative resemblance to how real-world worms have been known to propagate. We hope our simulation tool can contribute to teaching purposes and research purposes in the future.

As future work, the tool can be improved to propagate through more ways such as sequential scanning, routing scanning, hit-list scanning, selective attacks, etc. The interface can be improved by providing user-defined variable values of Ω, N, η, and controlling the speed of the time intervals. Further, for more advanced implementations, we could develop models for worm propagation in distributed networks, wireless networks [15] or introduce Machine Learning and Data Science tools to further improve our results. The developed simulation tool should be regularly updated to incorporate new changes as our knowledge about malware increases and new developments are made.

6. REFERENCES

[1] Jyotsna Krishnaswamy, Wormulator: Simulator for Rapidly Spreading Malware, in San Jose State University SJSU ScholarWorks, 2009. https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1068&context=etd_projects

[2] Wikipedia, Computer Worm, 2020. https://en.wikipedia.org/wiki/Computer_worm

[3] Veracode, Computer Worm. https://www.veracode.com/security/computer-worm

[4] Josh Fruhlinger, What is a computer worm? How this self-spreading malware wreaks havoc, https://www.csoonline.com/article/3429569/what-is-a-computer-worm-how-this-self-spreading-malware-wreaks-havoc.html

[5] SoftwareLab, What is a Computer Worm? https://softwarelab.org/what-is-a-computer-worm/

[6] Madihah Mohd Saudi, Kamaruzzaman Seman, Emran Mohd Tamil and Mohd Yamani Idna Idris, Worm Analysis through Computer Simulation (WAtCoS), The International Journal of Learning Annual Review, 2008. https://www.researchgate.net/publication/44262006_Worm_Analysis_through_Computer_Simulation_WAtCoS

[7] University of Central Florida, Performance Models of Computers and Networks, 2012. http://www.cs.ucf.edu/~czou/CDA6530-12/project-3.pdf

[8] Wikipedia, Discrete-event simulation model, 2020. https://en.wikipedia.org/wiki/Discrete-event_simulation

[9] TutorialsPoint, Discrete system simulation. https://www.tutorialspoint.com/modelling_and_simulation/modelling_and_simulation_discrete_system_simulation.htm

[10] Dmitry Kozyrev, Vladimir Rykov, Statistics and Simulation, 2018. https://www.researchgate.net/publication/325197422_On_Sensitivity_of_Steady-State_Probabilities_of_a_Cold_Redundant_System_to_the_Shapes_of_Life_and_Repair_Time_Distributions_of_Its_Elements

[11] Cliff C. Zou, Don Towsley, Weibo Gong, Modeling and Simulation Study of the Propagation and Defense of Internet Email Worm, University of Central Florida. http://cs.ucf.edu/~czou/research/emailWorm-TDSC.pdf

[12] Zesheng Chen, Worm Propagation Models, Georgia Institute of Technology. http://www.mathaware.org/mam/06/Chen.pdf

[13] Cliff Changchun Zou, Don Towsley, Weibo Gong, On the Performance of Internet Worm Scanning Strategies, Univ. Massachusetts, Amherst. http://www.eecs.ucf.edu/~czou/research/wormStrategy-techreport.pdf

[14] Markos Avlonitis, Emmanouil Magkos, Michalis Stefanidakis, Vassileios Chrissikopoulos, Exploring Scalability and Fast Spreading of Local Preference Worms via Gradient Models, 17th EICAR Annual Conference, 2008. https://www.researchgate.net/publication/253182633_Exploring_Scalability_and_Fast_Spreading_of_Local_Preference_Worms_via_Gradient_Models

[15] Farrah Kristel Batista, Angel Martín del Rey and Araceli Queiruga-Dios, A New Individual-Based Model to Simulate Malware Propagation in Wireless Sensor Networks, MDPI, 2020. https://doi.org/10.3390/math8030410