IMPLEMENTATION OF NETWORK BASED WORM INFECTION AND DETECTION

P.Doravelu Reddy
M Tech (Computer Science)
Department Of CSE
JNTUACEA, Ananthapur
Email id: doravelu.reddy@gmail.com

Dr C.Shoba Bindu M Tech, PhD
Associate Professor
Department Of CSE
JNTUACEA, Ananthapur
Email id: shobabindhu@gmail.com

ABSTRACT

Self-duplicating, self-propagating malicious codes known as computer worms spread themselves without any human interaction and launch the most destructive attacks against computer networks. In this paper, we have successfully detected the worm propagation characteristics of different and using permutation scanning to find the worms using branch process model to provide total number of scan that ensure the worm will eventually die out. Our strategy can effectively contain both fast scan worms and slow scan worms without knowing the worm signature in advance or needing to explicitly detect the worm. We would like to propose a statistical model for the spread of topology-aware worms and subsequently design mechanisms for automatic containment of such worms. We would also like to characterize the deviation of our proposed branching process model from the ideal sophistication epidemic model, assuming that the values of its rich set of parameters were available. Finally, we would like to port our worm containment schemes to edge routers and local routers and to evaluate the performance using real data from enterprise networks.

Index Terms – Statistical model, topology-aware worms, worm detection, worm infection, sophistication epidemic model.

I. INTRODUCTION

The first known worm was the Morris worm in 1988 November. Since then, New worms outbreaks have occurred periodically even though their mechanism of spreading was long well understood. So, the security threats and damaging disaster caused by network worms have increased dramatically. It has become apparent that no human intervention can react on timely enough to react to these types of attacks, and therefore automatic detection and prevention strategies against network worms are a necessity. On July 19, 2001, code-red worm (version 2) infected more than 250,000 hosts in just 9 hours [4]. Soon after, the Nimbda worm raged on the Internet. At 5:30 UTC on Saturday, January 25, 2003 the Slammer worm was released on the Internet [4]. At 5:33 it had achieve an aggregate scanning rate of over 55 million IP address scans per second. Within 10 minutes it had infected over 90% of the vulnerable population, around 75000 Microsoft SQL Servers. The goal of our research is to provide a model for the propagation of random scanning worms and the corresponding development of automatic containment mechanisms that prevent the spread of worms beyond their early stages. Our containment scheme is then extended to protect an enterprise network from a preference scanning worm. The host infected with random scanning worms finds and infects other vulnerable hosts by scanning a list of randomly generated IP addresses. Worms using other strategies to find vulnerable hosts to infect are not within the scope of this work. Some examples of non random-scanning worms are e-mail worms, peer-to-peer worms, and worms that search the local host for addresses to scan. Most modeling work concentrates on the relatively simple random-scanning worms, which scan the Internet either randomly or with bias toward local addresses in order to reach all vulnerable hosts. This strategy leaves a large footprint on the Internet (which reveals the worm's presence), and different infected hosts may end up scanning the same address repeatedly.

Our contributions are listed below:

- *Deterministic Epidemic Model* will detect the worm propagation only when large number of system is being affected.
- Inside the permutation scanning worms proposes a mathematical model that *precisely* characterizes a propagation patterns of the general permutation-scanning worms.
- The method we proposed will leads to find the neighbor systems details and to find the worm affected systems in the Network.
- Our automatic worm containment schemes effectively recover the IP address from its new one to exist IP address and Detect and Delete all the Worms available in the Network.

SEEK DIGITAL LIBRARY

II. RELATED WORK

Parbati Kumar manna et al[1] proposes a mathematical model that *precisely* characterizes the propagation patterns of the general permutation-scanning worms. The analytical framework captures the interactions among all infected hosts by a series of interdependent differential equations, which are then integrated into closed-form solutions that together present the overall worm behavior. Milan Vojnovi´c et al [2], the objective is to optimize the information spread with respect to minimizing the total number of samplings to reach a target fraction of the host population**.** David Litchfield et al [3] Of Next Generation Security Software discovered this underlying indexing service weakness in July 2002; Microsoft released a patch for the vulnerability before the vulnerability was publicly disclosed. Shigang Chen et al [4] propose a temporal rate-limit algorithm and a spatial rate-limit algorithm, which makes the speed of worm propagation configurable by the parameters of the defense system. Our “Implementation of Network Based Worm Infection and Detection” is based on statistical process model for characterizing the propagation of Internet worms.

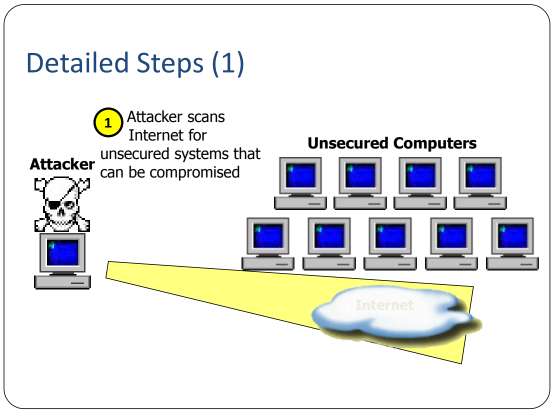
III. MODELING THE SPREAD OF ACTIVE WORMS THAT EMPLOY UNIFORM SCANNING

Our proposed automated worm containment strategy has the following steps:

- Let N be the total number of unique IP addresses that the host can contact in a containment cycle. At the beginning of each new containment cycle, set the counter that counts the number of unique IP addresses for each host to be zero.
- Increment counter for each host when it scans a new IP address.
- Hosts are thoroughly checked for infection at the end of the containment cycle (one by one to limit the disruption to the network).

The “heavy duty checking” could even include human intervention. The number of offending hosts is small, administrators should be able to take the machine offline and perform a thorough checking. The first step of the heavy-duty checking should be to follow a common security best-practice procedure. For example, One Should make sure that the antivirus software is up-to-date and is not disabled. One also needs to run a file integrity checker to make sure that the critical files are not modified, and no new executables are installed. After routine checking with all the available tools, an experienced system administrator should be able to make a final decision as to whether or not to let this machine be back.

In this paper We would like to propose a statistical model for the spread of topology-aware worms and subsequently design mechanisms for automatic containment of such worms. We would also like to characterize the deviation of our proposed branching process model from the ideal sophistication epidemic model. By spreading these worms it creates worm affected folders in all folder and subfolders in all the drives. Apart from that among all systems it should change the IP address for the systems which should be the vulnerable host. Changing the IP address creates lot of problems. Here in this paper we focus on how the worm spreading from one system to another system. After spreading that worm is it really change the IP address of the vulnerable host, if vulnerable host changes IP then in which way we reassign the old IP to that particular system.


Detailed Steps (1)

1
Attacker scans Internet for unsecured systems that can be compromised
Unsecured Computers
Attacker
Internet

Fig.1 How attacker scans the unsecured systems.

By the above diagrams we may illustrate that in which way attacker may scan the internet to find out the unsecured systems.

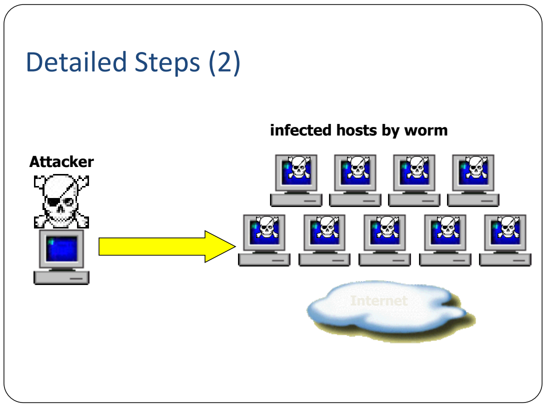

Detailed Steps (2)

Fig 2. Installation of worm program to vulnerable host.

By the above diagram it illustrates that after identifying the vulnerable hosts it installs the worms program into that vulnerable host and change the vulnerable hosts to infected hosts.

SEEK DIGITAL LIBRARY

A. *Propagation Model*

We now derive how i(t), a(t), u(t) and y(t) change over time t. Below we compute the amounts di(t), da(t), du(t), and dy(t) by which they change respectively over an infinitesimally small dt after time t. This will give us a set of differential equations that together characterize the propagation of 1-jump worms.

di(t): This, when multiplied by V, represents the total number of new infections generated during dt. Only effective (class u) hosts can hit new infections. Hence,

di(t) = u(t)*prob.

du(t): we will get effective hosts when we removed infected hosts from the vulnerable hosts. Hence,

$$du(t) = v(t)- \sum_{i=1}^{n} i(t).$$

And probabilities of attacking vulnerable hosts are

$$prob = \frac{u(t)}{N} . \qquad (1)$$

From the above analysis, we have

$$f_{hit} = r*dt*\frac{V}{N} \qquad (2)$$

$$du(t) = v(t) - \sum_{i=1}^{n} i(t) \qquad (3)$$

$$di(t) = u(t)*prob \qquad (4)$$

In this paper we are focusing that vulnerable hosts should be more than 16% of total number of hosts. Scan rate should be different for different range of infected hosts.

The propagation curves in Fig. 3, which are computed from the model (1)–(4) or collected from the simulations, demonstrate the topology-aware worms The number of infected hosts Vi(t). In the classical random-scanning worms, all infected hosts scan the Internet. The aggregate scan traffic peaks when all vulnerable hosts are infected. In the the topology-aware worms, only the active hosts scan. The number of active hosts, Va(t), can be much smaller than , which is evident from Fig, where the active curve is below the infected curve.

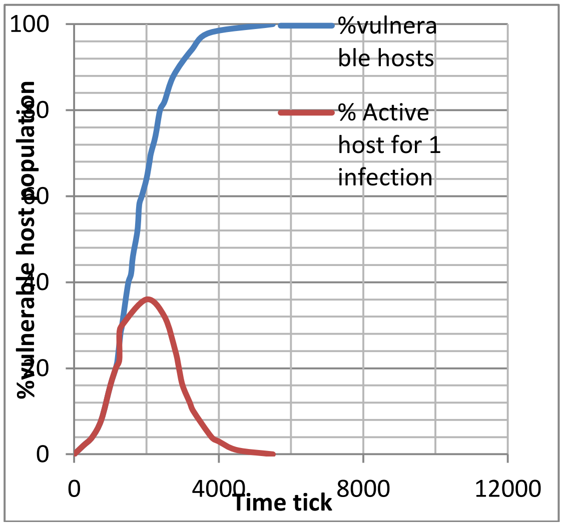


Fig 3 1 jump worms over time t

When the infected curve peaks at 100%, the active curve approaches to zero. That is, when all vulnerable hosts are infected, a random-scanning worm will reach the height of its scanning activity, whereas topology-aware worms will entirely conceal its presence and stay stealthy. The total volume of scan traffic by a permutation-scanning worm, which corresponds to the area under the active curve, is bounded. The total scan volume by a random-scanning worm, which corresponds to the area under its infected curve, will be much larger because the area is open-ended.

B. MODULE DESCRIPTION

i. WORM SPREAD

In Edge System:

Find all the neighboring systems connected to a particular LAN network. Transfer worm reading object to each host. In Neighbor Systems: The client receives the worm object. Spreading the worm object creates the victim files in all the folders & Sub folders. Change the IP address on each host dynamically.

ii. DETECTOR:

In Edge System:

Here in this we have designing and maintaining tables. Those are Host information table, Victim host table. Databases are connected to the table. Finding all machine information and assigning it to the table.

In Neighbor System:

The client receives the worm object. Detect the worm object and find the victim files in all folder and sub folders. Find the changed IP Address in the host information table.

iii. ANTI-WORM-SUB MODULES

The anti-worm traces the folders and finds out the worm object. The detected worm object is deleted from all folders. A thread is initialized to each client system which returns the IP address. Thread is executed continuously to find the change in IP address. Change in IP address is found by comparing last IP address and current IP address based upon database information.

C. PRACTICAL CONSIDERATION

In this section, we consider our model under real-world considerations; including congestion and bandwidth variability, patching and host crash, as well as delay of scan messages. The "Model" curves show the percentages of vulnerable hosts that are infected, active over time, respectively. These curves for i(t), a(t) are numerically computed from the analytical model. The "Simulation" curves are plotted using the averaged data collected from the simulator. As expected, for k=1, 2, 4 and 8, the curves from the model and the curves from the simulator completely overlap, which verifies the correctness of our model for k

SEEK DIGITAL LIBRARY

jump worms. The below diagram shows the infection curves (t) for a k jump worm under different k-values.

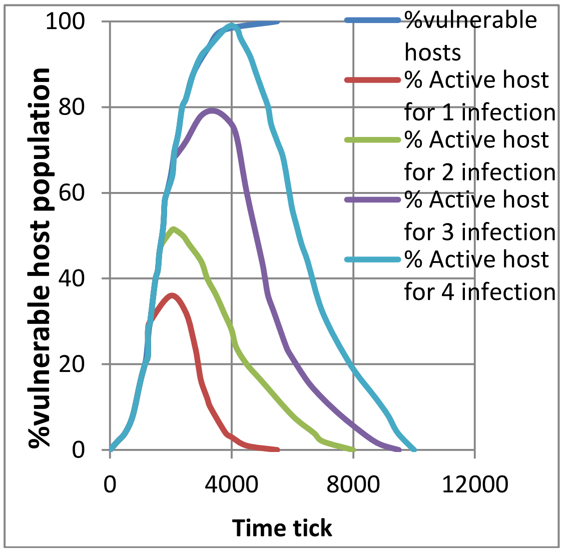


Fig4 Infected hosts for different k values.

The time taken to calculate Removal of infected hosts can be

$i_{removal}$ = di (t)*scan rate. (5)

By using the above formulae we can calculate time to be taken for removal of infected hosts. Here by attacking the worm it may change IP address of the infected host. To recover the IP address that means get back to the original IP address over time tick. By using simulation it should be shown below

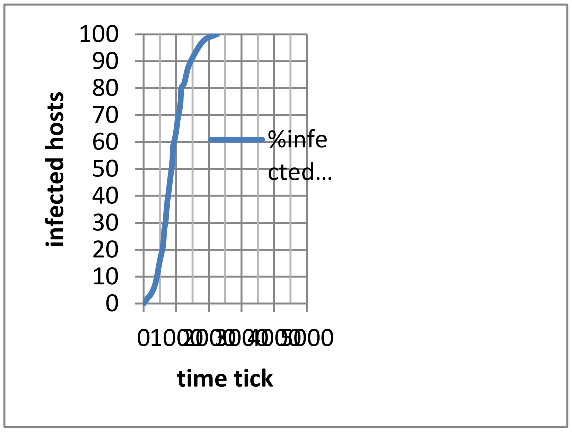


Fig5 recovery of Infected host IP address over time tick.

IV. Conclusion

In this paper, we have successfully detected the worm propagation characteristics of different and using permutation scanning to find the worms using branch process model to provide total number of scan that ensure the worm will eventually die out. Our strategy can effectively contain both fast scan worms and slow scan worms without knowing the worm signature in advance or needing to explicitly detect the worm. We would like to propose a statistical model for the spread of topology-aware worms and subsequently design mechanisms for automatic containment of such worms. We also show that our worm strategy, when used with traditional firewalls, can be deployed incrementally to provide worm containment for the local network and benefit the Internet. This model leads to the development of an automatic worm containment strategy that prevents the spread of a worm beyond its early stage. This model leads to find the neighbor systems details and to find the worm affected systems in the Network.The Advantages of Our automatic worm containment schemes effectively recover the IP address from its new one to exist IP address and Detect and Delete all the Worms available in the Network.

ACKNOWLEDGEMENT

The authors would like to express their cordial thanks to our parents, Dr. B.Eswara Reddy, Head Of Department, and Prof. A.Ananda Rao, Principal, JNTUACEA, Ananthapur, Andhra Pradesh..

REFERENCES

[1] Parbati Kumar Manna, *Member*, *IEEE*, Shigang Chen, and Sanjay Ranka, *Fellow*, *IEEE*, Inside the Permutation Scanning Worms: Propagation Modeling and Analysis

[2] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N.Weaver, “Inside the slammer worm,” in *Proc. IEEE Security Privacy*, Jul. 2003, vol. 1, no. 4, pp. 33–39.

[3] Z. Chen and C. Ji, “Optimal worm-scanning method using vulnerable host Distributions,” *Int. J. Security Netw.* vol. 2, no. 1/2, pp. 71–80, 2007, Special Issue on Computer and Network Security.

[4] M. Vojnovic, V. Gupta, T. Karagiannis, and C. Gkantsidis, “Sampling Strategies for epidemic-style information dissemination,” in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 1678–1686.

[5] J. Ma, G. M. Voelker, and S. Savage, “Self-stopping worms,” in *Proc. ACM Workshop Rapid Malcode (WORM)*, 2005, pp.12–21.[6] S. Schechter, J. Jung, and A. W. Berger, “Fast detection of scanning worm infections,” in *Proc. 7th Int. Symp. Recent Adv. Intrusion Detection*, Sep. 2004, pp. 59–81.

[6] David M. Nicol, Steve Hanna, Frank Stratton, William H. Sanders Information Trust Institute University of Illinois at Urbana-Champaign, Modeling and Analysis of Worm Defense Using Stochastic Activity Networks.

[7] Zesheng Chen Dept. of Electrical & Computer Engineering Georgia Institute of Technology Modeling the Spread of Active Worms.

SEEK DIGITAL LIBRARY