# On an Evaluation of Network Intrusion Dispersion

Surasak Sanguanpong[1] and Urupoj Kanlayasiri[2]

[1]Department of Computer Engineering, Kasetsart University, Bangkok, Thailand 10900
surasak.s@ku.ac.th
[2] Research Division, Office of the University Computer Services, Kasetsart University, Bangkok, Thailand 10900
urupoj.k@ku.ac.th

**Abstract.** The widespread of Internet worms is an issue of concern. A worm program tries to compromise systems in a large scale network. To handle the disaster proactively, it is very essential to evaluate the worm dispersion. We propose an evaluation model for the network intrusion dispersion on a spread of worms. The model scrutinizes important factors from a network infrastructure then calculates the dispersion value by using a rule-based fuzzy technique. These significant factors are generated from an analysis of trust relationship on nodes, the visibility of nodes, and the homogeneity of systems. The output of the model can be used to estimate the severity of the worm dispersion effectively.

## 1 Introduction

Worms are pieces of executable codes or programs that can automatically replicate themselves to machines by exploiting vulnerable services. When intruders discover system vulnerabilities, they try to gain access to the system and then propagate themselves to the other machines, thereby the spread of worms. From the history of worm, it has been clear since 1988 that the automatic propagation program can rapidly spread across network by exploiting the vulnerabilities of Sun Microsystems and VAX computers running variants of BSD UNIX [1]. The last few years, the dramatic increase of worm outbreaks occurred, namely, Code-Red and Sapphire/Slammer. Code-Red was infected by compromising Microsoft IIS web servers using the .ida vulnerability [3]. It launched 99 threads to generate the IP addresses of hosts randomly and then tried to compromise those machines with the same technique. However, the first version of Code-Red had limitations in random IP generator and its memory resident behavior.

The next version of Code-Red namely Code-Red II was released. It was much more dangerous than the prior version. Code-Red II was not memory resident; therefore, the rebooting an infected machine did not effect its operation [4]. This worm also improved the propagation mechanism to first check if the system has already infected, it then ignored to compromise again. The analysis papers of Code-Red I and Code-Red II are well-prepared in [3], [4], and [5]. The spread of Sapphire worm (also called Slammer) began to infect by exploiting buffer overflow vulnerability in Micro-

soft's SQL Server or Microsoft SQL Server Desktop Engine. It was regarded as the fastest worm in history [6]. Airline flight cancellations and ATM failures were the results of the worm dispersion. It performed the full scanning rate (over 55 million scans per second) to find the holes of stations. Notice that if Sapphire had carried a malicious payload and targeted a more popular service, the effects would likely have been more severe.

There are efforts to detect the behavior and signature of worms. GrIDS [7] is a well-known intrusion detection system developing to detect worms. It builds activity graphs to represent the host connections and then searches for the predefined pattern of intrusion from the graphs. It can handle a large number of hosts. To improve the effectiveness of GrIDS for detecting worms, the connection-history based anomaly detection [2] was proposed. The model emphasizes on the similarity of connection patterns, the causality of connection patterns, and the obsolete connections. The approach was described in different ways that it did not provide any specific patterns that should be searched for but defined a metrics that allowed the system to evaluate the likelihood that an observed pattern is malicious.

From the above researches and problem contexts, we realize that to quickly determine the spread of worm is very helpful to proactively handle with the situation of attack in a large scale. Therefore, the aim of this research is to propose a model to evaluate the network intrusion dispersion especially for the Internet worms. The remainder of the paper is organized as follows. Section 2 introduces the big picture and the basic concept of the model. Section 3 defines and analyzes the trust relationship that relates to the spread of worms. Also in Section 4, the openness and homogeneity of systems will be presented and analyzed. The evaluation of network intrusion dispersion using fuzzy technique is described in Section 5. Finally, Section 6 gives the conclusion and areas of future works.

## 2 The Model

The aim of this research is to analyze the significant factors effecting to the spread of worms in network. These factors will be utilized to estimate the intrusion dispersion.

### 2.1 Basic Concept

Dispersion relies on, but is not limited to, key entities: trust, openness, and homogeneity. *Trust* is a logical link between/among nodes for allowing or being authorized to perform actions. This relationship can be prepared in various means. For example, the relation "A trusts B" can be defined as the case when A allows B to access and write data on A's local disk. Trust is particularly a primitive way to support intruders to invade the system easier. The relationship can pass or inherit from one to the others. There have been suggestions that trust relationships should not be transitive because of security awareness. However, in a practical implementation, some configurations require a transitive trust.

*Openness* is a property that explains how the system can be seen or communicates with the others. This term describes the scopes of a system visibility. It is reasonable that the systems which are hidden from the scanning of intruders are safer from attacks than the world-visible machines. By the nature of worm propagation, after breaking-in to the victim it starts scanning for finding the next systems to be infected. The scanning tool looks for the specific vulnerability of the system that can make the worm to exploit and gain access. In addition, if targeted systems have the same hole, it will increase the possibility for worm to infect those machines in a large scale. We call this property as a *homogeneity*. Note that the homogeneity of the network is defined specifically for a particular attack.

### 2.2 Model Architecture

Three factors of interest are extracted from system and network configuration: trust, openness, and homogeneity.

Let $\delta$ be the function of dispersion evaluation. Then, the dispersion ($D$) is:

$$D = \delta(D_T, D_O, D_H) \quad (1)$$

where $D_T$, $D_O$, and $D_H$ are dispersion from trust, dispersion from openness, and dispersion from homogeneity, respectively. The overall architecture of the model depicts in Fig.1

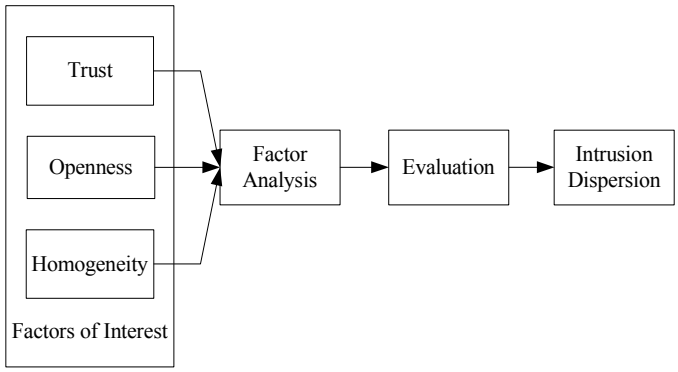


**Fig. 1.** Overall architecture of the model shows the basic components

The model starts with the extraction of factors of interest from the configuration of machines and network. Analysis part will analyze the factors. The final output for all processes is the intrusion dispersion that is calculated by the evaluation part.

## 3 Trust

Trust is a relationship representing action between a trustor and a trustee. The trustor allows the trustee to use, manipulate its resources, or influence the trustor's decision to use resources or services provided by the trustee. The trust relationship can be represented by a directed graph.

**Definition 1:** A *nondeterministic finite-state automaton M = (S, I, f, $s_o$, F)* consists of a set *S* of states, an input alphabet *I*, a transition function *f* that assists a set of states to each pair of state and input, a starting state $s_o$, and a subset *F* of *S* consisting of the final states.

We use the nondeterministic finite-state automaton *M* [8, 9] to describe the trust relationship of systems. The set of states *S* is a group of machines that have trust relationships. The function *f* represents the transition of worms to traverse from the node to infect the next node. By this definition, $s_0$ is the starting node that worm first exploits and *F* contains a set of possible attacked nodes. The input for function *f* is assumed to be a constant ($\lambda$).

$D_T$ (dispersion from trust) can be calculated by the following equation:

$$D_T = \frac{\sum_{i=1}^{n(S)} \left(n(F)_i - 1\right)}{\left(n(S) - 1\right)^2} \tag{2}$$

where $n(F)_i$ and $n(S)$ are the number of elements in set *F* with starting node *i* and the number of elements in set *S* respectively.

Fig. 2 and Table 1 demonstrate an example of trust relationship of systems. Using equation (2), $D_T$ is equal to 19/36 = 0.52.

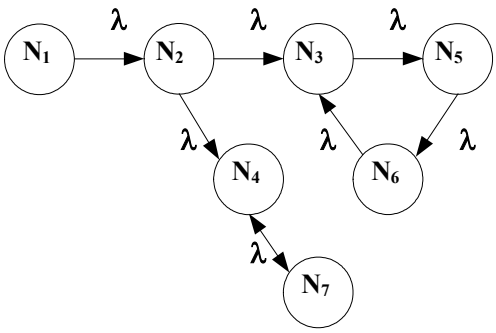


**Fig. 2.** A nondeterministic finite-state automaton represents trust relationships

**Table 1.** Possible attacked nodes.

| Starting node of attack ($s_o$) | Possible attacked nodes ($F$) | Number of possible infected nodes ($n(F)$-1) |
|---|---|---|
| $N_1$ | {$N_1$, $N_2$, $N_3$, $N_4$, $N_5$, $N_6$, $N_7$} | 6 |
| $N_2$ | {$N_2$, $N_3$, $N_4$, $N_5$, $N_6$, $N_7$} | 5 |
| $N_3$ | {$N_3$, $N_5$, $N_6$} | 2 |
| $N_4$ | {$N_4$, $N_7$} | 1 |
| $N_5$ | {$N_3$, $N_5$, $N_6$} | 2 |
| $N_6$ | {$N_3$, $N_5$, $N_6$} | 2 |
| $N_7$ | {$N_4$, $N_7$} | 1 |

## 4 Openness and Homogeneity

The visibility is a property that a host can be reached by the others. Machines in networks can be configured to hide themselves from attackers who are looking for them. The methods to achieve this property are such as firewall, network address translation, etc. By using these techniques, machines can be protected from the scanning of intruders. The desirable information for attackers are the availability of systems and the leakage of system information. Worms always choose an IP address randomly and try to establish a connection or send some packets to the victim. If they can reach that station, the scanning will be performed.

The dispersion from openness ($D_O$) can be defined by the following formula:

$$D_O = n(V) / n(S) \quad (3)$$

where $n(V)$ and $n(S)$ are the number of visible nodes and the number of all nodes in network respectively.

The example of nodes in the network with the property of openness demonstrates in Fig. 3. From the following figure, the $D_O$ value is 2/5 = 0.4

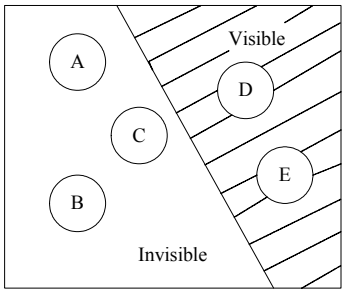


**Fig. 3.** The example of visibility of nodes in the network

The homogeneity of nodes is one of the important factors to determine the area of infection. When the worm can break-in the station, it usually uses the same vulnerability to invade the other hosts. Therefore, if many nodes have the same holes in operating systems or applications, the chance to be infected is increasing.

We can calculate the dispersion from homogeneity ($D_H$) by the following formula:

$$D_H = n(A) / n(S) \quad (4)$$

where $n(A)$ is the number of nodes that have the same hole A as the host that was first invaded by the worm and $n(S)$ is the number of all nodes in the network.

## 5 Evaluation

The calculation of dispersion utilizes a rule-based fuzzy technique that is suitable for defining and estimating the values of decision. Three factors ($D_T$, $D_O$, and $D_H$) are used to find the dispersion of worm. We employ rule-based fuzzy technique to aggregate the factors and calculate the dispersion. It starts with the fuzzification to assign membership values to inputs and output. Then the membership values of inputs are applied to 27 rules that were created by experts. Fig. 4 shows the example of the dispersion calculation when $D_H = 0.5$.

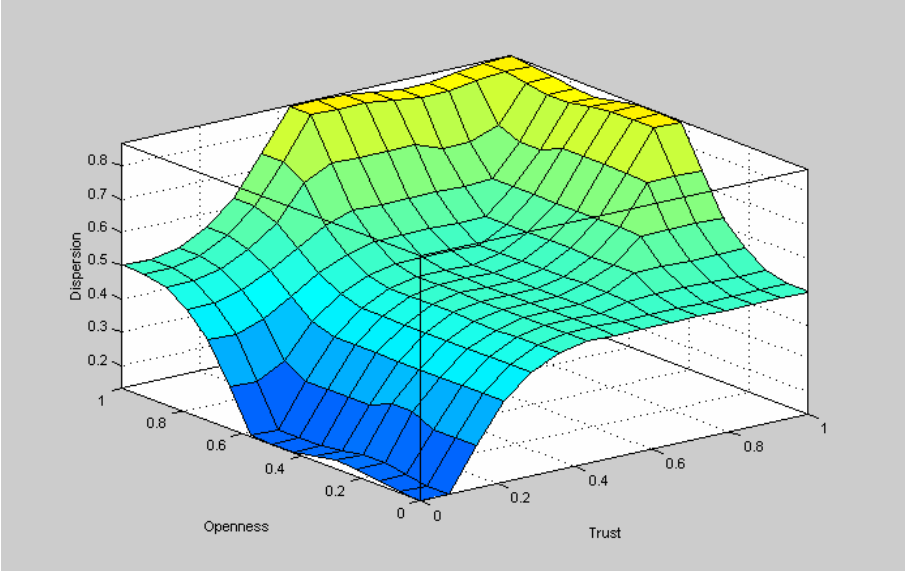


**Fig. 4.** The dispersion evaluation from fuzzy rule-based technique

## 6 Conclusion and Future Works

The paper describes the overall architecture and formal framework of the dispersion evaluation model for Internet worms. The significant factors are analyzed to find the possibility of worm to infect in a large scale network. After the first prototype is developed, the works will focus on the design of a general model to evaluate the dispersion that does not rely on system vulnerabilities and worm behaviors.

## Acknowledgements

Chalermkon Chongsanguan, Jittat Fakcharoenphol, Jitimon Keinduangjun, Punpiti Piamsa-nga, and referees made very helpful suggestions for which the authors are grateful. Thanks for the helps from Thanuwong Chaksupa and Sasakorn Nimviboon-som for paper preparation. This research is supported by Thailand Toray Science Foundation 2002 (TTSF).

## References


1. Spafford, E.: The Internet Worm: Crisis and Aftermath. Communication of the ACM, Vol. 32, No. 6 (1989) 678-687
2. Toth, T., Kruegel, C.: Connection-history based anomaly detection. Proceedings of the 2002 IEEE Workshop on Information Assurance and Security (2002) 30-35
3. Staniford, S., Paxon, V., Weaver, N.: How to 0wn the Internet in Your Spare Time. Proceedings of the 11th USENIX Security Symposium (2002)
4. Moore, D., Shannon, C.: Code-Red: a Case Study on the Spread and Victims of an Internet Worm. Proceedings of the 2002 ACM SICGOMM Internet Measurement Workshop (2002) 273-284
5. Moore, D., Shannon, C., Voelker, G., Savage, S.: Internet Quarantine: Requirements for Containing Self-Propagating Code. Proceedings of the 2003 IEEE Infocom Conference (2003)
6. Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., Weaver, N.: The Spread of the Sapphire/Slammer Worm. Technical Report, http://www.caida.org/analysis/security/sapphire/ (2003)
7. Staniford-Chen, S., Cheung, S., Crawford, R., Dilger, M., Frank, J., Hoagland, J., Levitt, K., Wee, C., Yip, R., Zerkle, D.: GrIDS-A Graph based Intrusion Detection System for Large Networks. Proceedings of the 19th National Information Systems Security Conference
8. Rosen, K.: Discrete Mathematics and Its Applications. 3rd edn. McGraw-Hill New York (1995)
9. Linz, P.: An Introduction to Formal Languages and Automata. 2nd edn. D.C.Health and Company Massachusetts (1996)