Cyber Weather Forecasting: Forecasting Unknown Internet Worms Using Randomness Analysis

Hyundo Park, Sung-Oh Jung, Heejo Lee, Hoh In

Cyber Weather Forecasting: Forecasting Unknown Internet Worms Using Randomness Analysis *

Hyundo Park[1], Sung-Oh David Jung[2], Heejo Lee[1], and Hoh Peter In[1]

[1] Korea University, Seoul, Korea, {hyundo95, heejo, hoh_in}@korea.ac.kr
[2] Trinity International University, IL, USA, {zsjung}@tiu.edu

Abstract. Since early responses are crucial to reduce the damage from unknown Internet attacks, our first consideration while developing a defense mechanism can be on time efficiency and observing (and predicting) the change of network statuses, even at the sacrifice of accuracy. In the recent security field, it is an earnest desire that a new mechanism to predict unknown future Internet attacks needs to be developed. This motivates us to study forecasting toward future Internet atacks, which is referred to as CWF (Cyber Weather Forecasting). In this paper, in order to show that the principle of CWF can be realized in the real-world, we propose a forecasting mechanism called FORE (FOrecasting using REgression analysis) through the real-time analysis of the randomness in the network traffic. FORE responds against unknown worms 1.8 times faster than the early detection mechanism, named ADUR (Anomaly Detection Using Randomness check), that can detect the worm when only one percent of total number of vulnerable hosts are infected. Furthermore, FORE can give us timely information about the process of the change of the current network situation. Evaluation results demonstrate the prediction efficiency of the proposed mechanism, including the ability to predict worm behaviors starting from 0.03 percent infection. To our best knowledge, this is the first study to achieve the prediction of future Internet attacks.

Keywords: Forecasting, Internet worm, Randomness check, Regression analysis, Reliability check

1 Introduction

In the recently security field, various intelligent Internet attacks are being developed and they cause fatal damages on the Internet. In cases of previous fatal Internet incidents, we experienced that an Internet worm has an extremely high destructive force and devastates the Internet. Since the destructive power of a worm is caused by its propagation speed, faster worm propagation techniques

* This work was partially supported by Seoul City R&BD program WR080951 and the National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (2009-0086140)

have been used in an Internet attack. For example, when a bot master constructs a botnet with compromising other hosts, a bot master can use a worm propagation technique. As a result, it is crucial to cope with a worm at an early stage of a worm propagation.

So far, many researchers have been made a commitment to develop worm detection approaches. Though their efforts, we can decide whether our network is under a worm propagation situation or not. In general, approaches to detect a worm are classified into two categories: signature based and anomaly based. In general, a signature based worm detection approach is widely used on most systems. The reason is that these approaches have the advantage of low false-positive rates. However, they cannot handle an unknown worm properly. Unlike a signature based worm detection approach, an anomaly based worm detection approach detects an unknown worm [1, 4, 5, 9, 10, 14–16]. They enable us to recognize even an unknown attack by examining the network situation whether it is normal or not. In order to do that, the most of detection approaches evaluate particular metrics as numeric values, representing the network situation. Unfortunately, when the worm just began its existence on the Internet, most anomaly worm detection approaches cannot detect it because they need at least a certain amount of attack traffic to decide whether the network situation satisfies the condition of an abnormal network situation or not. Moreover, the result of an anomaly detection mechanism is based on a binary decision, 'true' or 'false' (whether the network is under an attack or not). With the only result of an anomaly detection approach, we cannot observe the changing process of the network situation under a worm propagation. Observing the change of the network situation, from being benign to being a worm propagation recognized by an anomaly detection approach, will be very useful to cope with a worm. In other words, if we can observe the process of the change of the network situation before recognizing as a worm propagation (e.g., after one hour, the situation will turn out to be in worm propagation), it will be very useful to cope with a worm distinctly.

The concept of forecasting is to estimate future statuses by examining and analyzing current information [3]. There are several examples of forecasting such as a sales/stock forecasting in the business or a weather forecasting in the meteorology. Thus, we have a hypothesis that forecasting future statuses of the network situation by analyzing the current status of the network situation can be accomplishable. Forecasting future trends of the network situation is the concept of *CWF* (Cyber Weather Forecasting). If we analyze current trends of a worm propagation, we can estimate future trends of a worm propagation. The reason is that the number of infected hosts and attack packets generated by them is continuously increased but not decreased as time passes, as shown in cases of Cod-Red and Slammer [2, 13, 17, 18]. In their studies, we divide the process of a worm propagation into three stages as follows;

– *SP (Starting Point)*: this stage is very early stage of a worm propagation and the symptom caused by a worm propagation is a subtle sign.

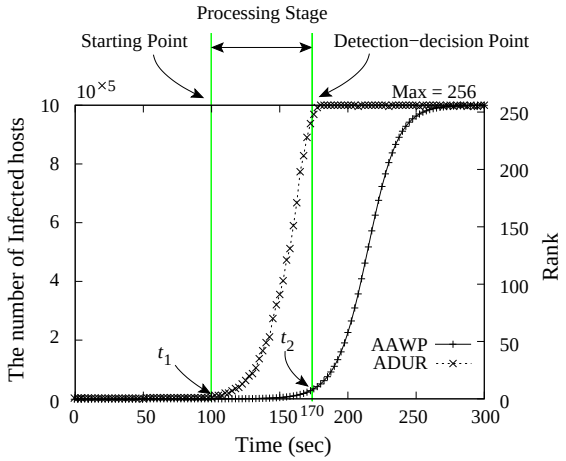

Fig. 1. Three phases of an early worm detection mechanism; Starting Point, Processing Stage and Detection-decision Point.

- *PS (Processing Stage)*: a worm propagation is in progress and the symptom caused by a worm propagation is getting clearer but it is still a subtle sign.
- *DP (Detection-decision Point)*: the symptom caused by a worm propagation is clear enough to recognize a worm propagation situation by an anomaly worm detection approach.

Fig. 1 shows that ADUR (Anomaly Detection Using Randomness check), proposed by Park *et al.* as one of early worm detection approaches based on the threshold, detects the worm at an early stage of the worm propagation [9, 10]. In Fig. 1, the worm propagation is based on the AAWP (the Analytical Active Worm Propagation) model, one of the popular models for the Internet scale worm propagation [2]. As shown in Fig. 1, ADUR has three phases (*SP*, *PS* and *DP*) under a worm propagation. It is pretty much the same as other worm detection approaches based on the threshold. In Fig. 1, the rank value, estimated from ADUR, is maintained in low values under the benign network situation or the very early worm propagation stage (before *SP*). But the symptom of a worm propagation becomes clearer under *PS* (after *SP* until *DP*) and the value increases more and more. Finally, when the symptom caused by a worm propagation becomes clear, the value goes over the threshold at *DP*. When our network situation is under *SP* or *PS*, *DP* will be the future network situation. Nevertheless, *CWF* forecasts *DP* with analyzing the current trend of the change of the network traffic under *PS*. To forecast *DP*, *CWF* decides whether the current network situation is *SP* or not. If our network situation is *SP*, our network situation turns to *PS* and *CWF* forecasts *DP* with the time from the current time until *DP* as shown in Fig. 1. In Fig. 1, at time t_1, the current network situation

is *SP* and the *CWF* forecasts the future network situation with $\Delta t = t_2 - t_1$. As time passed, t_1 will be closed to t_2 and *CWF* still forecasts with Δt until *DP*.

In this paper, we propose a new conceptual model of *CWF*, and we show that *CWF* can be realized in the real-world with applying statistical theories. To show that *CWF* can be accomplished in the real-world, we provide the value, the prediction of attack situation by analyzing the values taken from ADUR. The evaluation results of our forecasting approach, called FORE (Forecasting using REgression analysis), show that FORE can predict the Internet worm 1.8 times faster than ADUR does, irrespective of the number of vulnerable hosts or the worm scan rate. Furthermore, FORE continuously forecasts Δt under *PS*. This result shows that predicting and alerting an unknown attack is possible and this predicted information will be invaluable for a security agencies.

This paper is organized as follows. In Section 2, we give an in-depth account of *CWF*. In Section 3 and Section 4, we explain the ADUR and FORE mechanism. Section 5 demonstrates the effectiveness of the FORE with the evaluation results. Finally, we conclude with a summary of contribution and discuss future works.

2 Cyber Weather Forecasting

In this paper, to help us better understand *CWF*, we explain the limited *CWF* within which predicts a worm propagation.

We can get information of a current network situation related to a worm propagation using an early worm detection mechanism. Nevertheless, we always have curiosities toward the future network situation, such like how and when the current network situation will be changed to the worm propagation network situation in the future. Unfortunately, since most early worm detection mechanisms merely estimate the network situation with the binary decision, such as 'true' or 'false', we cannot investigate how the network situation is being changed during the period from triggering the worm propagation to detecting it. As a result, developing the new model which can continuously report a possibility of the worm epidemic before when the early worm detection mechanism detects the worm is earnestly desirable, even at the sacrifice of accuracy.

In order to satisfy our desires, we propose the model of *CWF* in this paper. *CWF* is the conceptual model that forecasts the change of the future network situation. *CWF* forecasts when the current situation will be clearly changed to the worm epidemic situation and reports the changing process of the network situation during the period from triggering the worm to detecting it. Therefore, *CWF* helps us to get the information of the future network situations (reporting the trend of worm propagation under *PS* and forecasting the time of *DP* with Δt)

In the view points of forecasting the future Internet worm, Nazario *et al.* proposed a mathematical model for a vulnerability's "wormability", the potential for the use of the vulnerability in worm propagation [8]. The model provides time intervals between the publication of the vulnerability and the current time; and the model measures the possibility of several vulnerability announcements in the

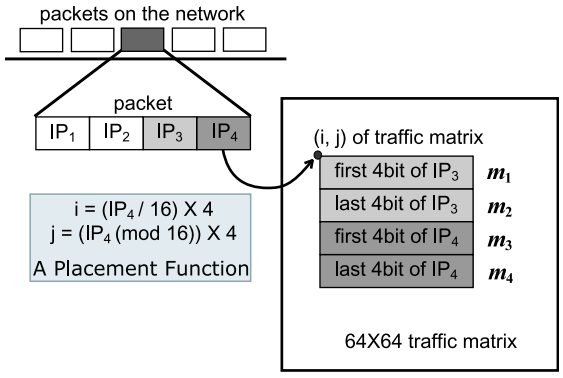

Fig. 2. Constructing matrix M by mapping a packet to submatrix m.

near future along with explanations of why some vulnerabilities are inherently not "wormable". On the other hand, Sanguanpong *et al.* proposed the approach to minimize the damage due to worm infection in enterprise networks [12]. The authors predict the number of infected nodes by fuzzy decision, thus benefiting to the study of worm behaviors.

Unfortunately, these studies cannot be included in *CWF*. Since previous studies predict the worm propagation using theoretic models, related to previous worm trends and worm propagation behaviors, and it is absolutely impossible to find the point when the unknown worm propagation will be triggered on the Internet, these studies are useful in analyzing already known worms only.

To realize *CWF* in the real-world, two types of information which are used to predict the worm should be reported to us as follows:

- *BP* (Branch Point) : reporting the starting time of forecasting the worm (i.e., *SP*).
- *RT* (Remaining Time) : reporting the remaining time from the current to the future worm epidemic situation (i.e., Δt).

In Fig. 1, if the worm propagation is triggered on the Internet, *CWF* senses the change of the network situation with the tiny symptom of the worm propagation, even at the sacrifice of accuracy. Thus, *CWF* reports *BP* (as *SP*) and *RT* (as Δt)

3 ADUR (Anomaly Detection using Randomness check) : The Context

Since we propose a forecasting model called FORE which analyzes the randomness of the network traffic, we first explain ADUR in concrete. ADUR, proposed by Park *et al.* [9, 10], is the anomaly worm detection mechanism using randomness checks of the network traffic. In order to do that, ADUR uses the matrix

operation after constructing the traffic matrix. In order to check the randomness of the network traffic, ADUR uses the well-known binary matrix rank test, proposed by Marsaglia *et al.* [6, 7].

Since one prominent characteristic of Internet worms is a random selection of subsequent targets, ADUR examines the random distribution of destination addresses in the network traffic. ADUR consists of four steps: 1) traffic matrix construction, 2) XOR operation, 3) rank calculation, and 4) randomness checkup. In order to obtain more accurate attack information such as a randomness of either entering or departing worm traffic from the network, ADUR classifies traffic data into two categories based on their direction: incoming and outgoing. ADUR constructs the traffic matrix, a binary square matrix, to check a randomness of traffic, because checking for randomness of a binary square matrix is easily accomplished by measuring the rank of the matrix. As well, ADUR uses a XOR operation to diminish the effect of normal traffic on rank values.

First, ADUR constructs the traffic matrix as Fig. 2. To represent the traffic information (the binary information of third and fourth octets of IP address) on the traffic matrix, ADUR finds the location, i and j of the traffic matrix, with the fourth octet of IP address of a packet and overwrites the binary information of third and fourth of octets of IP address of a packet on the submatrix (4×4) in the traffic matrix (64×64). As a result, a randomness of traffic can be properly expressed on the matrix. In previous works, ADUR considers the network environment as IPv4 when ADUR constructs the traffic matrix. To apply ADUR to the IPv6 network environment, we should adopt a new method to find the location, i and j of the traffic matrix, and the function to find the location should satisfy three conditions (it should be easy to compute the index for any given IP address, difficult to find an IP address that maps to a given index value, and difficult to spoof the IP address without the index value being changed). As a result, under the IPv6 network environment, we can adopt a cryptographic hash function to find the location. In this paper, we consider the network environment as IPv4.

Second, since the legitimate traffic can reduce accuracy of the worm detection, a simple XOR operation is very efficiently used to remove the legitimate traffic on the traffic matrix.

$$R(M'_t) = R(M_t \oplus M_{t-1}) \tag{1}$$

where M_t is the traffic matrix at time t. Eq. (1) presents the XOR operation on a sequence of matrices. Let M'_t denote the result of the XOR operation of two consecutive matrices, M_t and M_{t-1}. Then, the XOR operation eventually removes most of legitimate traffic in the traffic matrix M'_t because legitimate traffic lives longer than one time unit so the portion of legitimate traffic is eliminated by the XOR operation. This effectiveness of the XOR operation will be constant even when the network state goes the busy state (such like the traffic volume increases highly) caused by huge legitimate users (flash crowds) [11].

Third, the common dimension of the row space and the column space of a matrix is called the rank. A straightforward method to compute a rank of a

matrix is to count the number of non-zero rows after applying the Gaussian elimination to the matrix. The rank of the matrix is equal to the number of non-zero rows (or equivalently, the number of leading 1's) on the matrix. In Eq. (1), $R(M'_t)$ is the rank value of M'_t.

Finally, when the worm propagates on the Internet, the traffic has randomness. Thus, ADUR detects the worm with the rank value of the traffic matrix. The rank value of the 64×64 random matrix exceeds 60 with the probability which is greater then 99.999%. This implies that, if the 64×64 binary matrix is a random matrix then the rank has a high probability of being greater than 60. To summarize, the rank of the matrix can be used to determine the randomness of distribution of the elements on the matrix. This 64×64 matrix will undoubtedly be used when a /24 network is monitored [10]. When we monitor the network larger than a /24 network, we should enlarge the size of matrix [9]. However, in this paper, we use a 64×64 binary matrix as early research to predict unknown worms.

ADUR effectively detects the Internet worm at an early stage of worm propagation. It is shown that ADUR is highly sensitive so that the worm epidemic can be detectable quickly, *e.g.* three times earlier than the infection of 90% vulnerable hosts [9,10] which is evaluated from AAWP model [2]. And when ADUR detects the worm epidemic, the number of infected hosts was only one percent of the total number of vulnerable hosts. It means that ADUR can detect the worm epidemic before the fast spread phase of the worm propagation, represented by Zou *et al.* [18].

4 FORE (FOrecasting using REgression analysis) : The Proposed Model

Here we propose a forecasting model called FORE. FORE finds *BP* when the worm emerges from the Internet and predicts *RT*.

FORE consists of three steps: 1) time series analysis, 2) linear regression analysis, and 3) reliability analysis. In these three steps, step 1) and step 2) are used to estimate *RT*, and step 3) is used to find *BP*. FORE does not use out-rank values (of out-coming traffic matrix of ADUR) but in-rank values (of in-coming traffic matrix of ADUR) because out-rank values are only useful to detect the worms originating from the monitored network.

4.1 Time Series Analysis

In time series analysis, a moving average can be used for smoothing the scattered in-rank data. Eq. (2) shows the moving average of rank values

$$\bar{R}_i = \left(\sum_{j=i-t}^{i} R_j \right) / t \tag{2}$$

where R_i, t, and $\bar{R}_i$ are the in-rank value at time i, the time interval to calculate average, and the average of in-rank values from $i-t$ to i, respectively.

4.2 Linear Regression Analysis

Linear regression analysis yields the equation of the fitted line of the in-rank values of the incoming traffic matrix. As seen in Eq. (3), the equation is used to predict RT. The best feature of worm propagation behavior is that the number of infected hosts does not decrease but increases in case that the self-propagation worm spreads over the Internet [2,13,17,18]. Therefore, the equation of the fitted line of the in-rank values is used for calculating the in-rank values at a specific time and vice versa.

$$R_i = \alpha + \beta T_i \tag{3}$$

where R_i and T_i are the in-rank value at time tick i, and the time at i, respectively.

$$\beta = \left(\sum_{j=i-t}^{i} \left(R_j - \bar{R}\right)\left(T_j - \bar{T}\right) \right) / \left(\sum_{j=i-t}^{i} T_j - \bar{T} \right) \tag{4}$$

$$\alpha = \bar{R} - \beta T_i \tag{5}$$

where $\bar{R}$ and T are an average of results of the moving average of in-rank values on time interval t and the average of time on time interval t, respectively. We can get α and β using Eq. (5) and Eq. (4) at time tick i. With Eq. (3) based on α and β, FORE predicts when the rank value goes over the threshold. In this paper, since the threshold of ADUR is 60, R_i of Eq. (3) will be 60 so FORE can forecast RT (as the value of $\Delta t = T_i - t_i$ where t_i is the current time).

4.3 Reliability Analysis

In the reliability analysis, the coefficient of determination measures the reliability of the equation of the fitted line, and measuring the reliability represents the trust level of predicting the detection decision point when ADUR detects the worm. Added to that, the reliability is used to find the BP to start predicting the worm.

$$C_i = \left(\sum_{j=i-t}^{i} \left(\widehat{R_j} - \bar{R}\right)^2 \right) / \left(\sum_{j=i-t}^{i} \left(R_j - \bar{R}\right)^2 \right) \tag{6}$$

where C_i and $\widehat{R_i}$ are the reliability of the fitted line and the in-rank value on the fitted line after applying the moving average of original in-rank values, respectively. As shown in Fig. 3, in Stage 1 (before BP), since rank values merely oscillate, the reliability of the similarity between the fitted line (obtained using Eq. (3)) and in-rank values is very low (not over 0.8 as shown in Fig. 3). In contrast, in Stage 2 (during PS), since the number of infected hosts and

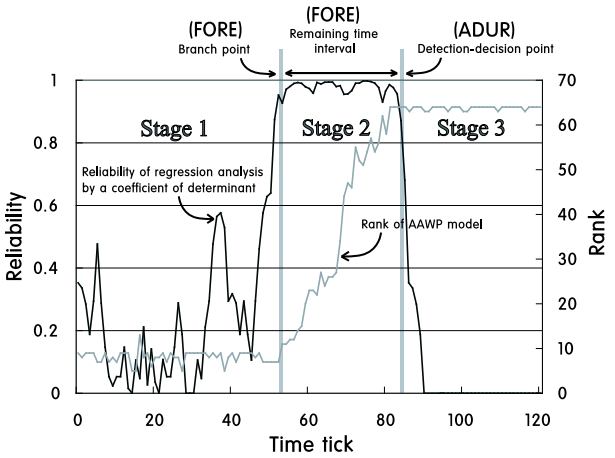

Fig. 3. The estimated branch point using reliability analysis.

attack packets generated by the worm is continuously increased but not decreased as time passes, the reliability of the similarity between the fitted line and in-rank values is very high (over 0.8 as shown in Fig. 3). According to this phenomenon, FORE finds BP using this difference among the reliability values (as C_i in Eq. (6)).

All these analyses are performed using the in-rank values within the contiguous time interval; the time interval exists between the previous time and the current time. The previous time must be continuously selected while reflecting network environment.

In this section, first, FORE decides whether the current time is BP or not using the reliability analysis. If the current time is BP, FORE finds the fitted line and predicts RT using regression analysis. In the following section, we show evaluation results of FORE where FORE predicts the worm effectively. Thus, it will be confirmed that CWF can be accomplishable.

5 Evaluation Results

FORE is evaluated using the real worm traffic by generating as many as the number of infected hosts, being estimated using the AAWP model. In the model, it is assumed that the number of vulnerable hosts is one million, and the monitored network is a /24 network. In order to evaluate the efficiency and reliability of FORE, time-series analysis and regression analysis are employed using real traffic dump data; and the data is comprised of packets captured on the backbone of a university campus network in July 29, 2004, a /16 campus network traffic trace where the average number of clients and packets per second are 132 and 1,847 during the time spam of 187 seconds. And the most busy /24 network trace for background traffic is used to evaluate FORE during 180 seconds. The

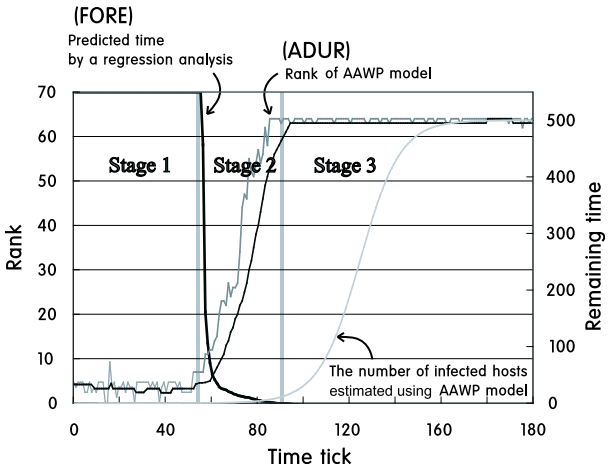

Fig. 4. Trajectory of the remaining time predicted by FORE.

traffic matrix is constructed with one second time unit. The time unit of the moving average and the regression analysis is 10 seconds.

As shown in Fig 3, the worm propagation is divided by three parts: the Stage 1, the Stage 2 and the Stage 3.

- Stage 1 : from the start time to *BP* (*i.e.*, before *SP* in Fig. 1)
- Stage 2 : from the branch point to *DP* (*i.e.*, during *PS* in Fig. 1)
- Stage 3 : after *DP* (*i.e.*, after *DP* in Fig. 1)

FORE estimates in-rank values using reliability analysis. As a result, FORE senses the tiny change of the network traffic by the worm, and decides the end of the Stage 1 and the start of the Stage 2 with *BP* which is drown by the reliability analysis. Following the start of the Stage 2, ADUR checks whether the in-rank value goes over the threshold or not. If the in-rank value goes over the threshold, ADUR decides the end of the Stage 2 and the start of the Stage 3 using randomness checks. In our evaluations, the threshold of reliability is 0.8 drown by experimental results. As shown in Fig. 3, FORE decides whether the current network situation is *BP* (as *SP* in Fig. 1) or not.

Fig. 4 plots in-rank values of ADUR, results after applying the moving average to in-rank values of ADUR, and remaining time values estimated by FORE. In Fig. 4, 'Predicted time by a regression analysis' shows the trajectory of the predicted *RT* as time passes. Since the moving average requires a contiguous time period (as 10 seconds in Fig. 4) to calculate the average of in-rank values, *RT* value does not present the correct *RT* at the early stage of Stage 2. But, after the early stage of Stage 2 (as after 63 time tick in Fig. 4), *RT* presents a correct value.

In Fig. 4, FORE starts to predict RT when only 0.03 percent of the total vulnerable hosts are infected. At this time, the number of infected hosts is merely

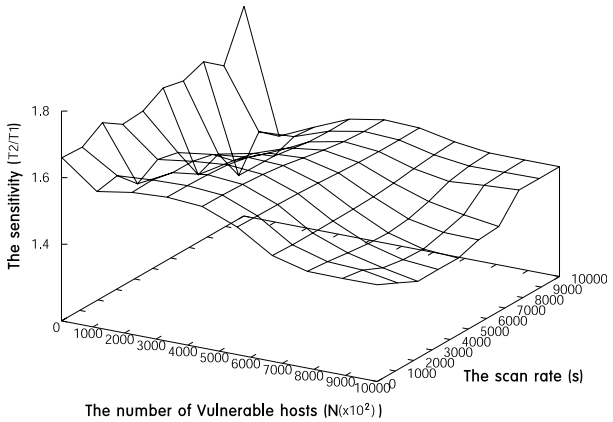

Fig. 5. Comparison of the sensitivity of FORE comparing with ADUR.

306 of the total number of vulnerable hosts. Furthermore, ADUR detects the worm when 1 percent of total vulnerable hosts are infected. In the same manner, the number of infected hosts is 10306 of total vulnerable hosts when ADUR detects the worm.

Fig. 5 shows the FORE's sensitivity, measured under the worm spreading state which is generated by AAWP model, where T_1, T_2 , s, and N are BP by FORE, DP by ADUR, the worm scan rate per second, and the number of vulnerable hosts on the Internet, respectively. In Fig. 5, the faster the worm scan rate, the higher sensitivity FORE has. Likewise, the smaller the number of vulnerable hosts make FORE more sensitive. Fig. 5 shows that FORE predicts the Internet worm 1.8 times faster than ADUR does, irrespective of the number of vulnerable hosts or the worm scan rate.

6 Conclusions

For the purpose of realizing *CWF*, the FORE mechanism is proposed and evaluated for predicting worm epidemics with analyzing dynamics of a randomness in the network traffic. The evaluation results show that FORE responds 1.8 times faster than the early detection mechanism (e.g., ADUR) against unknown worms. As these results address, this study shows the incorporation of forecasting into the detection of unknown worms (e.g., zero-day worms). Furthermore, using FORE, we can observe the change of network situation before when the early detection mechanism (e.g., ADUR) detects the worm propagation situation with binary decision, 'true' or 'false'. To the best of our knowledge, this is a new direction to predict the future worm epidemics based on the current state of the monitored network traffic. In addition to the worm prediction, we are sure that FORE can be applied to predict many other Internet attacks, if the attack changes the situation of the network traffic with time, unlike a normal network

situation. And if the attack detection mechanism estimates the network situation with the value and the threshold, we are sure that FORE can be applicable for predicting future attacks in an earlier stage than any detection mechanism.

References

[1] Berk, V., R.S.Gray, Bakos, G.: Flowscan: Using sensor networks and data fusion for early detection of active worms. SPIE AeroSense 5071, 92–104 (2003)

[2] Chen, Z., Gao, L., Kwiat, K.: Modeling the spread of active worms. In: IEEE INFOCOM (2003)

[3] InvestorWords.com: What is forecast? definition and meaning, `http://www.investorwords.com/2038/forecast.html`

[4] Jung, J., Paxon, V., Berger, A., Balakrishnan, H.: Fast portscan detection using sequential hypothesis testing. In: IEEE Symp. on Security and Privacy. pp. 211–225 (2004)

[5] Leckie, C., Kotagiri, R.: A probabilistic approach to detecting network scans. In: IEEE Network Operations and Management Symposium (NOMS) (Apr 2002)

[6] Marsaglia, G.: Diehard: a battery of tests of randomness, `http://stat.fsu.edu/~geo/diehard.html`

[7] Marsaglia, G., Tsay, L.H.: Matrices and the structure of random number sequences. Linear algebra and its applications 67, 147–156 (1985)

[8] Nazario, J., Ptacek, T., Song, D.: Wormability: A description for vulnerabilities. Arbor Networks (Oct 2004)

[9] Park, H., KIM, H., Lee, H.: Is early warning of an imminent worm epidemic possible? IEEE Network 23(5), 14–20 (Oct 2009)

[10] Park, H., Lee, H., Kim, H.: Detecting unknown worms using randomness check. IEICE Trans. on Communications E90-B(4), 894–903 (Apr 2007)

[11] Park, H., Li, P., Gao, D., Lee, H., Deng, R.H.: Distinguishing between FE and DDoS using randomness check. In: Information Security Conference(ISC2008), LNCS. vol. 5222, pp. 131–145 (2008)

[12] Sanguanpong, S., Kanlayasiri, U.: Worm damage minimization in enterprise networks. Int'l Journal of Human–Computer Studies 65(1), 3–16 (Jan 2007)

[13] Staniford, S., Paxson, V., Weaver, N.: How to own the internet in your spare time. USENIX Security Symposium pp. 149–169 (Aug 2002)

[14] Tong, X., Wang, Z.: A novel anomaly detection algorithm and prewarning technology of unknown worms. Communications in Computer and Information Science, 163, 164–171 (2011)

[15] Whyte, D., Kranakis, E., Oorschot, P.: DNS-based detection of scanning worms in an enterprise network. In: Network and Distributed System Security Symposium (NDSS) (2004)

[16] Whyte, D., Oorschot, P., Kranakis, E.: ARP-based detection of scanning worms within an enterprise network. In: Annual Computer Security Applications Conference (ACSAC). pp. 5–9 (2005)

[17] Wu, J., Vangala, S., Gao, L., Kwiat, K.: An effient architecture and algorithm for detecting worms with various scan techniques. NDSS (Feb 2004)

[18] Zou, C., Gong, W., Towsley, D., Gao, L.: The monitoring and early detection of internet worms. IEEE/ACM Transaction on networking 13(5), 961–974 (Oct 2005)