경량화 에이전트를 이용한 확장성 있는 분산 웜 탐지 및 방지 모델

(A Scalable Distributed Worm Detection and Prevention Model using Lightweight Agent)

박 연 희 [†] 김 종 욱 [††]
(Park Yeonhee) (Kim Jonguk)

이 성 욱 [†††] 김 철 민 [††††]
(Lee Seong-uck) (Kim Cholmin)

우 즈 만 [††] 홍 만 표 [†††††]
(Usman Tariq) (Hong Manpyo)

요 약 웜은 사람의 개입 없이 취약점이 존재하는 네트워크 서비스에 대한 공격을 시행하고 사용자가 원치 않는 패킷을 복사 및 전파하는 악성코드이다. 기존의 웜 탐지 기법은 주로 시그너쳐 기반의 방식이 주를 이루었으나 조기 탐지의 한계로 인해 최근에는 웜 전파의 행동 특성을 감지하는 방식이 각광 받고 있다. 본 논문에서는 웜 행동 주기와 감염 체인으로 대표되는 웜의 행위적 특성을 탐지하고 대응할 수 있는 분산 웜 탐지 및 방지 방법을 제안하고, 제안된 탐지 및 방지 모델 적용 시 웜의 감염 속도가 감소되는 현상을 시뮬레이션을 통해 증명한다. 제안하는 웜 탐지 모델은 규모가 큰 시그너쳐 데이타베이스가 필요하지 않을 뿐더러 컴퓨팅 파워가 비교적 적게 소요되므로, 개인용 컴퓨터 뿐 아니라 유비쿼터스와 모바일 환경과 같이 개별 기기가 낮은 컴퓨팅 파워를 가지는 상황에도 적합하다.

키워드 : 분산 웜 탐지, 조기 탐지, 악성 코드

Abstract A worm is a malware that propagates quickly from host to host without any human intervention. Need of early worm detection has changed research paradigm from signature based worm detection to the behavioral based detection. To increase effectiveness of proposed solution, in this paper we present mechanism of detection and prevention of worm in distributed fashion. Furthermore, to minimize the worm destruction; upon worm detection we propagate the possible attack alert to neighboring nodes in secure and organized manner. Considering worm behavior, our proposed mechanism detects worm cycles and infection chains to detect the sudden change in network performance. And our model neither needs to maintain a huge database of signatures nor needs to have too much computing power, that is why it is very light and simple. So, our proposed scheme is suitable for the ubiquitous environment. Simulation results illustrate better detection and prevention which leads to the reduction of infection rate.

Key words : Distributed Worm Detection, Early Detection, Malicious Code

1. 서 론

A computer worm is a self-replicating malicious program. In the WAN environment, worms are able to propagate in a very short time and cause the entire networks to be paralyzed [1-3]. Human response is not quick enough to detect fast-scanning worms.

Signature based virus detection mechanisms are commonly used by many security software vendors. Previously researchers monitor network packets to detect and protect from malicious code automatically. Proliferation of packets with the same payload is considered to be malicious and blocked [4, 5]. But such schemes require relatively high threshold to reduce a false positive. Hence, it is poor to realize early detection.

Considering limitations of previous research works,

· 이 논문은 제34회 추계학술대회에서 '경량화 에이전트를 이용한 확장성 있는 분산 웜 탐지 및 방지 모델'의 제목으로 발표된 논문을 확장한 것임
· This research is supported by Foundation of ubiquitous computing and networking project (UCN) Project, the Ministry of Knowledge Economy (MKE) 21st Century Frontier R&D Program in Korea and a result of subproject UCN 08B3-B1-30S.

† 학생회원 : 아주대학교 정보통신공학과
piaoyj@ajou.ac.kr
†† 비 회 원 : 아주대학교 정보통신공학과
kju@ajou.ac.kr
usman@ajou.ac.kr
††† 비 회 원 : 신구대학 인터넷정보과 교수
suleeip@shingu.ac.kr
†††† 비 회 원 : 시스온칩 부설 무선통신연구소
cmkim@sysonchip.co.kr
††††† 종신회원 : 아주대학교 정보통신공학과 교수
mphong@ajou.ac.kr
논문접수 : 2007년 12월 14일
심사완료 : 2008년 3월 26일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제5호(2008.7)

many researchers tried to find a better early worm detection methods by analyzing macroscopic propagation characteristics of worms [6,7].

In this paper, we present collaborative packet marking and attack alert reporting scheme for early worm detection and prevention. Simulation results showed that our solution is very effective to suppress the spread of worms.

2. Background and Related Works

Wider and frequent nature of worm propagation catches a lot of researcher's attention. N.Weaver et al. [2] is recommended for the taxonomy of worms and a review of the worm terminology.

Early research focus was on threshold based worm detection [4]. The threshold-based detection method encounters high false positive when we set high threshold value. High false positive problem changed research focus to analyze the characteristics of self-replication to detect worms.

Intrusion detection systems (IDS) are deployed to discover worm intrusions. Signature-based IDS [8,9] can detect well-known worms easily but it is hard for them to identify unknown worm.

Our previous research of this paper was Distributed Worm Detection Model based on packet marking. This model detects worm cycle and infection chain which is main behavior features of worms [10]. It does not need signatures with worm packet contents or traffic profiles. This model is highly scalable and feasible because of its distributed reacting mechanism and low processing overhead. Each host informs the possibility of worm propagation to recipients using packet marking technique in Distributed Worm Detection Model. If a marking counter is larger than the threshold, the host considers this packet as worm packet and stops further transmission of packet. However, in this situation, blocking of packet transmission just happens in the host where its marking counter is greater than the threshold, and parent of the host still deliveries packets. Thus our previous method is not perfect and our new approach is complementary to this effort.

3. The Proposed Idea

Proposed model neither increases the network traffic nor requires any special processing servers. Each host conducts the necessary information of path which is associated with that host. Each detection model requires minute information to activate itself; such as infection tree depth and only check of IP header identification field in received messages which create negligible overhead on network and system resources.

3.1 Model 1 - Backward Reporting

Figure 1 shows the scenario of our model. Let's assume that the host 'A' has been infected by worm.

Step1. The worm in host 'A' becomes active. It replicates itself and propagates to B, C, F, and G. Host 'A' embeds a marking counter value 1 and a report value (1 or 0) in fragmented IP header identification field before delivering the packets to next hop.

Step2. After receiving the marking embedded packet at host B, C, F, and G, the receiving nodes will try to connect with other hosts; and the marking counter in packet will be increased to inform other hosts that it is a continuous connection. For example, Host H has received packet with a marking counter 2 from B, tries to connect with host M, N, O, and P. In order to inform M, N, O, and P that it is a continuous connection, host H sends a packet with increased marking counter 3. Thus the marking counter plays a role to indicate the number of continuous connections among hosts. When one host gets packets with several different marking counters, we only considers the maximum value.

Step3. If the marking counter in received packets is greater than the predefined threshold, these packets are considered as suspicious and the host stops delivering these packets. Thus the packet is considered as a worm if the depth of infection tree is over a certain value.

Step4. Each host which has received the worm's replication decides whether it should report packet information to its parent by analyzing its report value in received packets. As a chain length increases the total amount of scanning in a host increases, all hosts which are infected from one

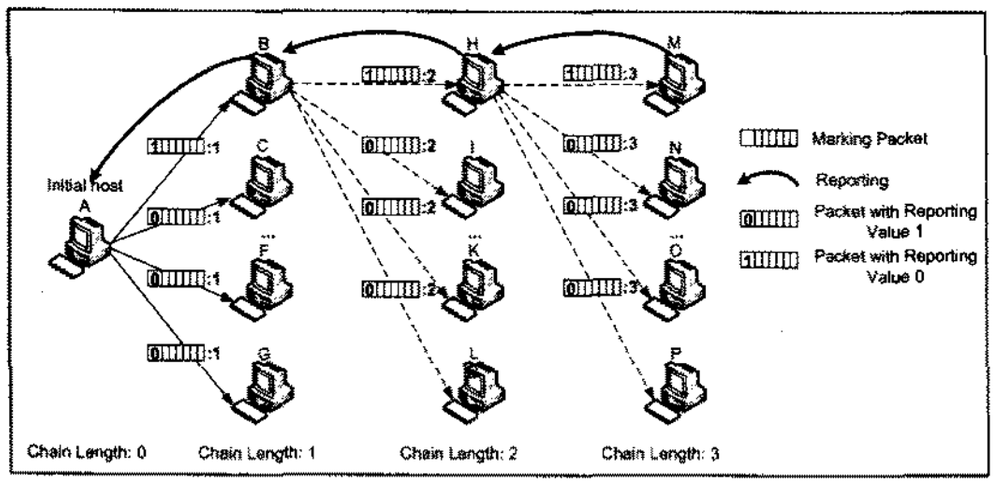

Fig. 1 Infection chain using packet marking and backward reporting scheme

host have to report to one parent node, which leads to the increment of traffic. Hence, reporting ratio is set to decide whether a node should report to its parent or not. The host attaches report value to IP header Identification field before delivering it. The report value depends upon reporting ratio.

For example, let us assume every host's reporting ratio is 50%. It means 50% of hosts having the same parent will report and notify their parent to stop further packet transmission. Surely this report is sent when the marking counter in the infected host is greater than the threshold and supposed to be suspicious.

However, even if the report-rate is 100%, number of reporting packets will not be large enough to exert influence on the network traffic.

Step5. After receiving the report, host stops the delivery of packets and reports to its parents until the parent is discovered.

3.2 Model 2 - Backward & Forward Reporting

The neighbor host is a host in which our detection and prevention system is installed in. Model 2 uses not only backward reporting mentioned in model 1 but also forward reporting, letting more hosts turn into the immune state. In other words, in model 2, amount of immune hosts has been increased which is faster than model 1.

Step1 - Step5. They have same actions with Model 1.

Step6. Each host which received the worm's replication will report to its neighboring hosts and let neighbors turn into the immune state. We assume that a host knows all of its neighbors. This feature will allow any host to inform its neighboring nodes about worm infection proactively which will leads to diminish the infection rate. As mentioned in section 3.1, the reporting packet has very minimal over head on network.

Model 2 tries to let the agent-installed hosts turn into immune state in advance by using the forward reporting. Therefore, vulnerable hosts without an agent have lower probability that they can be infected from the host with an agent.

4. Simulation

We virtually construct a worm propagation environment. Table 1 shows the properties of each host.

Our simulation has been done with agent-rate, birth-rate, scan-count, report-rate, and neighbor-count which define in Table 2. Figure 2 is used to simulate worm propagation process.

In our simulation we made 50,000 hosts in which few nodes stage and draw a comparison between each model. X-axis of each graph denotes the number of stages and Y-axis denotes the number of infected hosts.

Figure 3 shows the simulation with sim.1 parameters described in Table 3 and displays the worm propagation stages of the our previous paper

Table 1 The property of each host

Property	Description
Parent ID	The value which can identify host's parent
Infected	Whether the host is infected or not
Immuned	Whether the host is in a immune state or not
Agent installed	Whether our agent is installed or not
Received alert	Whether the host received report or not
Marking counter	The marking counter value

Table 2 Definitions of Agent-rate, Birth-rate, Scan-count, Report-rate and Neighbor-count

Parameter	Description
Agent-rate	The proportion of the hosts with our agent in all hosts
Birth-rate	The probability that a worm will be activated in a host.
Scan-count	The number of hosts that an infected host scans in a time unit.
Report-rate	The probability that a host which has agent will report to its parent.
Neighbor-count	The number of neighbors of a host with our agent.

Table 3 Simulation parameters

Properties	Sim.1	Sim.2	Sim.3
Total number of hosts	50,000	50,000	50,000
Agent-rate	75%	50%	50%
Birth-rate	100%	100%	100%
Scan-count	5	5	5
Marking counter threshold	4	4	2
Neighbor-count	5, 25	5, 25	5, 25
Report-rate	100%	100%	100%

[10] (from now on, we name the model of previous paper 'no reporting'), as well as Model 1 and Model 2 in this paper. From our results comparing the no reporting and forward reporting in our research, we can confirm the intuitive result that forward reporting scheme can lower the peak value. We also can see that the spread of worm is suppressed and the number of infected hosts is decreased. That is because, using no reporting, though hosts already had our agent, if the host which has our agent is not in an immune state, it will keep delivering packets until it turns into an immune state. In our idea, once a worm packet is detected, the host will report to parents and neighbors, so as to let more hosts turn into an immune state earlier. The spread of worms can be controlled in the early stage.

Figure 4 shows the simulation with Sim.2 parameters in Table 3. As we can see in Figure 3 and Figure 4, the prevention effect of our Model 2 with agent-rate 50% does not make much difference to the result of no reporting scheme with agent-rate 75%. When agent-rate is 75%(Figure 3), the peak of number of infected hosts by no reporting is about 25,200, when agent-rate is 50%(Figure 4), the peak value in Model 2 with 25 neighbors is about 27,700.

Figure 5 shows the simulation with Sim.3. From our result comparing 2 and 4 of marking counter threshold values in Figure 4 and Figure 5, we can confirm the result that raising the threshold value for the purpose of decreasing false positive can also prevent hosts from infecting. The greater the

Procedure Propagation Stage

Variables
'H' is a set of hosts in the simulation environment. H[i] means i^{th} host of H.
Each host has 5 properties:
'infected' : Boolean variable.
'immuned' : Boolean variable.
'propagation_count' : Integer variable, means chain length.
'report' : Boolean variable, means whether report or not.
'receive_alert' : Boolean variable.
't' is threshold

Initialize
1 **while** FOREVER
2 **for** i←1 to |H|
3 **if** (H[i].infected = TRUE && H[i].immuned = TRUE)
4 **then if** (H[i].propagation_count >= t)
5 **then** H[H[i].neighbors].infected←FALSE
6 **if** (H[i].report = TRUE)
7 **then** H[i].infected ←FALSE
8 H[H[i].infected_from].receive_alert←TRUE
9 **else if**(H[i].report = FALSE)
10 H[i].infected←FALSE
11 H[H[i].infected_from].receive_alert←FALSE
12 **goto 2**
13 **else if** (H[i].propagation_count < t && H[i].receive_alert = TRUE)
14 **then** H[i].infected←FALSE
15 H[H[i].infected_from].receive_alert←TRUE
16 **goto 3**
17 **else if** (H[i].propagation_count < t && H[i].receive_alert = FALSE)
18 **call** WormCycle(H[i])

Function WormCycle

Input
'H[a]' is an infected host
Variables
'V' is a set of 'H' which can be infected by H[a].
'b' is the birth rate of the worm.
'j' is a random real number between 0 to 1.

1 **for** i←1 to |V|
2 **if** (j<b)
3 **then** V[i].infected←TRUE
4 **if** (V[i].propagation_count < H[a].propagation_count + 1)
5 **then** V[i].propagation_count←H[a].propagation_count + 1
6 V[i].infected_from←H[a]

Fig. 2 Algorithm for simulation model

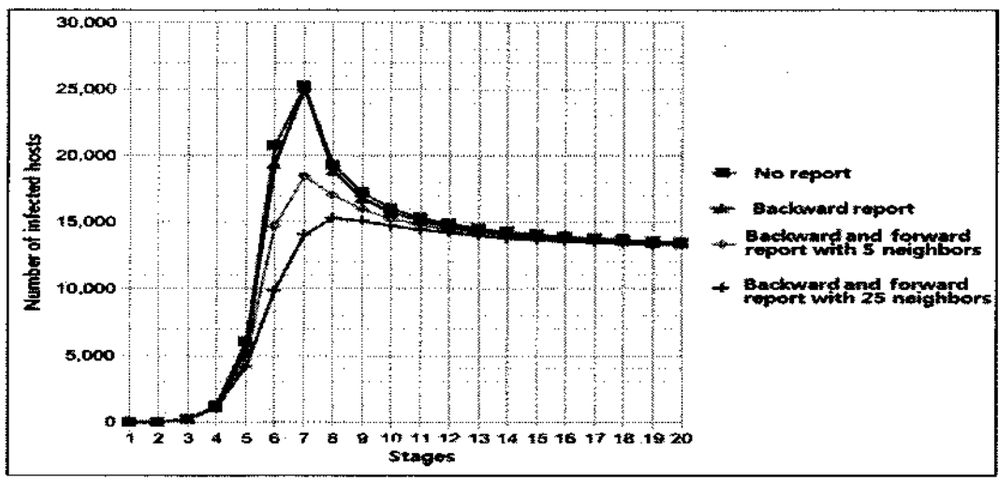

Fig. 3 Worm propagation stages with agent-rate 75%, threshold 4

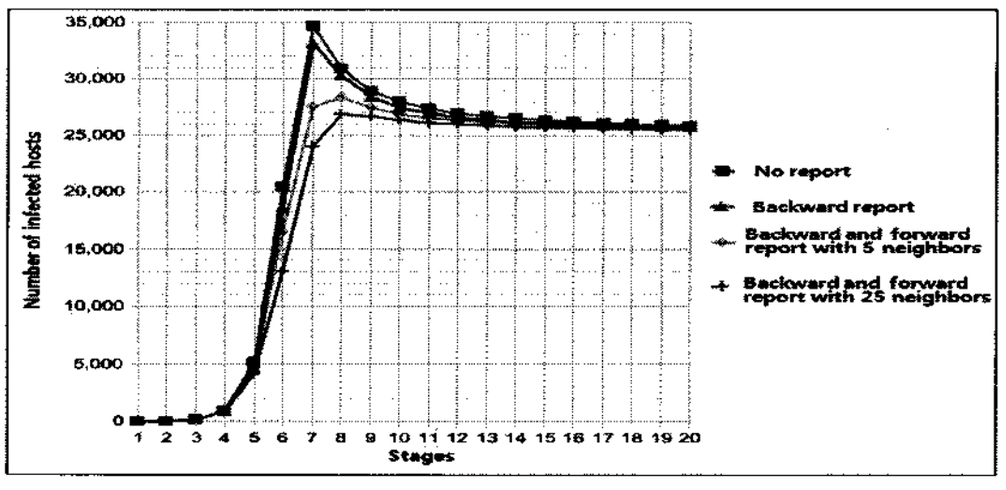

Fig. 4 Worm propagation stages with agent-rate 50%, threshold 4

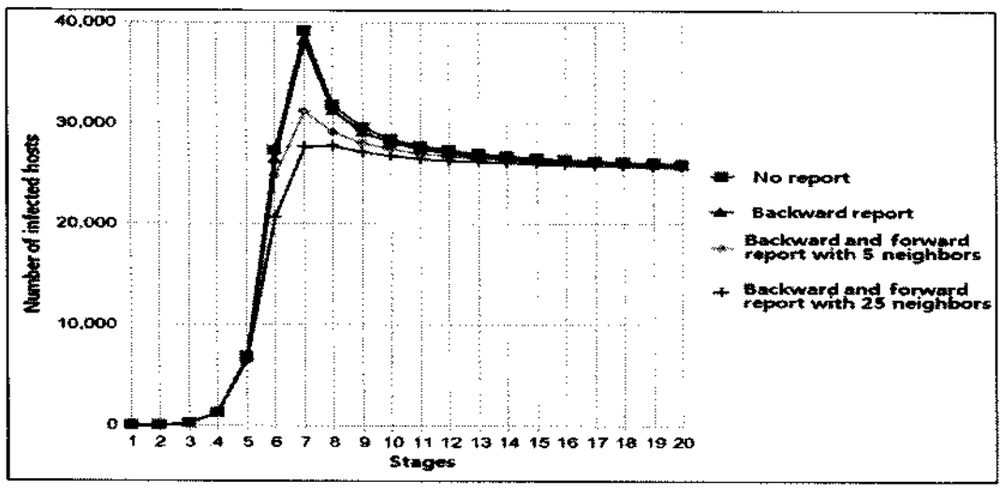

Fig. 5 Worm propagation stages with agent-rate 50%, threshold 2

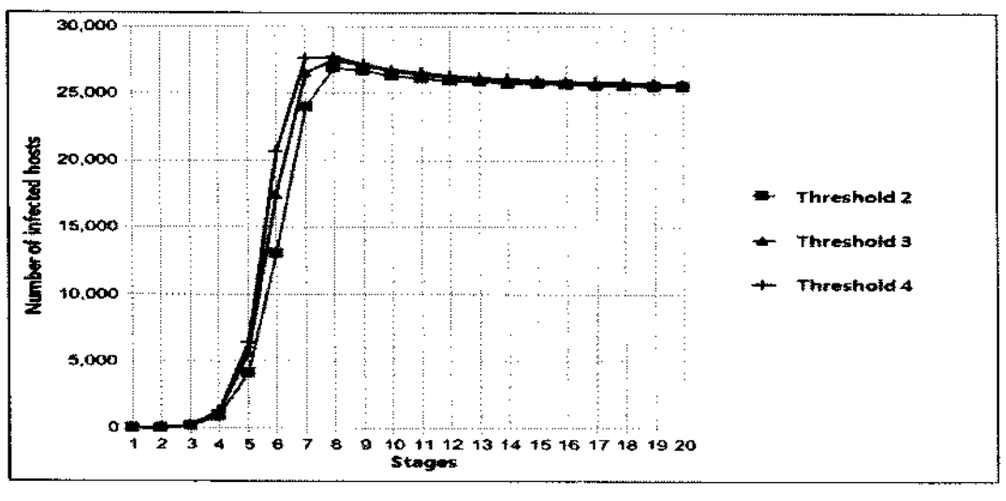

Fig. 6 Worm propagation stages with agent-rate 50%, threshold 2, 3, 4

threshold is, the higher the peak of the number of infected hosts becomes with no reporting. But in Model 2, as the threshold goes higher, the peak value becomes little higher. With no reporting, the difference of the number of infected host between the threshold 2 and 4 is about 5000, which is 10% of all hosts. Using backward and forward reporting method, the difference is about 1000, which is 2% of total.

In order to examine false positive of our model, we set different threshold values. Figure 6 shows the simulation with threshold values 2, 3, and 4. Although we set a little higher threshold to decrease false positive, the maximum number of infected hosts does not change much.

5. Conclusion

In this paper, we have presented Distributed Worm Detection and Prevention Model, a novel approach to defend against worm attacks. Each host only analyzes the necessary information of the infection path associated with that host by using packet marking and reporting scheme. A marking counter is used to transfer the number of continuous connections among hosts. The received packets at host who's marking counter greater than the pre defined threshold, are considered as suspicious. Upon worm detection, node will backward and forward, an alert report to parent and neighboring node which helps to increase the immunity of the whole system and the number of infected hosts would be decreased even if worm propagation is continued.

References

[1] David Moore, Vern Paxson, Stefan savage, Colleen Shannon, Stuart Staniford, and Nicholas Weaver, "Inside the Slammer worm," IEEE security and Privacy, July 2003.

[2] Nicholas Weaver, Vern Paxson, Stuart Staniford and Robert Cunningham, "A taxonomy of computer worms," In WORM'03 October 27, 2003.

[3] Stuart E.Schechter, Jaeyeon Jung, Arthur W. Berger, "Fast Detection of Scanning Worm Infections," Recent Advances in Intrusion Detection, 7th International Symposium, RAID 2004, Sophia, France, September 15-17, 2004.

[4] H.A.Kim, B.Karp, "Autograph: Toward Autumated, Distributed Worm Signature Detection," In Proc. Of 13th Usenix Security Symposium, August, 2004.

[5] Nicholas Weaver, Stuart Staniford, Vern Paxson, "Very Fast Containment of Scanning Worms," Proc. Of the 13th Usenix Security Conference, 2004.

[6] Y.Xie, V.Sekar, D.Maltz, M.K.Reiter, and H.Zhang, "Worm Origin Identification Using Random Moonwalks," In Proc. Of IEEE Symposium on Security and Privacy, 2005.

[7] Y.Al-Hammadi, C.Leckie, "Anomaly Detection for Internet Worms," 2005 9th IFIP/IEEE International Symposium, Intergrated Network Management, 2005.

[8] Martin Roesch, "Snort-lightweight intrusion detection for networks," In USENIX Large Installation Systems Administration Conference, Seattle, WA, USA, November 1999.

[9] Vern Paxson, "Bro: a system for detecting network intruders in real-time," Computer Networks, 1999.

[10] Kangsan Lee, Cholmin Kim, Seong-uck Lee, Manpyu Hong, "Macroscopic Treatment to Unknown Malicious Mobile Codes," Journal of KISS, 2006.

[11] A.S.Savage, D.Wetherall, A.Karlin, and T.Anderson, "Practical network support for IP traceback," In Proc. Of the 2000 ACM SIGCOMM conference, August 2000.