http://dx.doi.org/10.7236/JIIBC.2013.13.1.71
JIIBC 2013-1-10

Deterlab 환경에서 Earlybird를 이용한 웜 탐지와 Snort 연동을 통한 웜 확산 차단

Worm Detection and Containment using Earlybird and Snort on Deterlab

이형윤*, 황성운**, 안병구***

Hyeong-Yun Lee, Seong-Oun Hwang, Beongku An

요 약 웜이란 시스템의 취약점을 탐색하고 취약한 시스템을 공격하여 훼손시키는 독립형 프로그램으로서, 네트워크를 통하여 자신을 복제하고 확산한다. 본 논문에서는 웜 탐지 및 차단 방법을 연구하였다. 먼저 가상 시뮬레이션 테스트베드인 Deterlab 환경에서 Codered Ⅱ 웜 트래픽을 발생시켰다. 이 트래픽을 Earlybird를 이용하여 의심스러운 부분을 식별한 후, Wireshark를 통해 분석하여 Snort 규칙을 작성하였다. 다음으로 Codered Ⅱ 웜 트래픽에, 앞에서 작성된 Snort 규칙을 적용함으로써, 생성된 로그 파일의 확인을 통해, 정상적으로 웜 탐지가 이루어짐을 확인할 수 있었다.

Abstract A computer worm is a standalone malware computer program that probes and exploits vulnerabilities of systems. It replicates and spreads itself to other computers via networks. In this paper, we study how to detect and prevent worms. First, we generated Codered Ⅱ traffic on the emulated testbed called Deterlab. Then we identified dubious parts using Earlybird and wrote down Snort rules using Wireshark. Finally, by applying the Snort rules to the traffic, we could confirmed that worm detection was successfully done.

Key Words : Worm, Detection, Earlybird, Snort, IDS(Intrusion Detection System)

Ⅰ. 서 론

최근 고속 광대역 네트워크를 통하여 멀티미디어 서비스를 비롯한 다양한 서비스가 가능하게 되었다. 반면에, 바이러스, 웜(Worm), 스파이웨어, 애드웨어 등 네트워크를 통한 악의적인 공격 또한 과거에 비해 많이 증가하고 있다. 네트워크를 통한 여러 가지 공격 중 가장 심각한 피해를 초래하는 것은 웜이다. 웜이란 사용자의 개입 없이 네트워크를 통해 스스로 자신의 복사본 또는 변형체를 퍼뜨려 취약점이 있는 네트워크, 시스템 또는 서비스를 공격하는 악성 코드를 말한다. 웜은 감염된 시스템의 파일을 지우거나 변형시켜 중요한 정보를 훼손시키거나 이메일을 통해 유출시키는 등의 피해를 준다.

기존의 웜은 이메일의 첨부파일을 실행시키는 것처럼, 사용자가 감염된 파일을 실행하거나, 감염된 매체를 시스템에 연결하는 등 사용자의 개입에 의해 전파되었으므로 전파 속도에 한계가 있었고 피해 범위가 넓지 않았다. 이에 반해 최근 웜의 경우 임의의 IP 주소 대역을 가진

*준회원, 홍익대학교 컴퓨터정보통신공학과
**정회원, 홍익대학교 컴퓨터정보통신공학과
***종신회원, 홍익대학교 컴퓨터정보통신공학과
접수일자 : 2012년 12월 12일, 수정완료 : 2013년 1월 21일
게재확정일자 : 2013년 2월 8일

Received: 12 December 2012 / Revised: 21 January 2013 / Accepted: 8 February 2013
**Corresponding Author: sohwang@hongik.ac.kr
Dept. of Computer & Information Communications Engineering, Hongik University, Korea

시스템들에 지속적으로 자신의 복사본을 전파함으로써, 짧은 시간 안에 아주 넓은 범위에 걸쳐서 피해를 준다. 최근 피해 사례를 살펴보면, 2001년에 발생하여 14시간 동안 약 359,000 대의 컴퓨터를 감염시켰던 CodeRed, 2003년 발생하여 10분 만에 약 75,000 대의 컴퓨터를 감염시켰던 Slammer 등이 있다. 이와 같이 최신의 웜은 사용자가 미처 대응하기 전, 짧은 시간 안에 네트워크에 전파되므로 실시간으로 웜을 탐지하고 대응할 수 있는 시스템이 필요하다[1].

이러한 웜의 탐지를 위해 많은 연구가 진행되고 있지만, 대부분의 연구가 수동적으로 웜 또는 악성코드를 분석하여 고유 시그너처 (signature)를 생성하고 이를 이용하는 방법에 중점을 두고 있다. 하지만, 이 방법은 사람이 수동적으로 패킷을 검사해야 하므로 탐지에 소요되는 시간이 길어 신속히 대응할 수 없는 문제가 있다[2].

또한, 랜덤한 네트워크 대역으로 확산되는 웜의 특성을 이용한 탐지에 관한 연구는 미비한 실정이다. 이 방법은 트래픽 특성 및 주소기반을 이용한 탐지 방법으로 위의 수동 탐지에 비해 정확도가 떨어지지만 빠른 시간에 효율적인 탐지가 가능하다.

따라서 본 논문에서는 기존의 시그너처 기반 탐지 방법을 기반으로, 트래픽 특성 및 주소 기반을 이용한 탐지 방법을 제안한다. 구체적으로 Deterlab의 가상 시뮬레이션 환경에서 CoderedⅡ 웜 트래픽을 발생시키고, Earlybird 도구를 이용하여 웜 시그너처를 생성한 다음, 이를 오픈 소스 침입탐지시스템인 Snort에 연동하여 웜 확산을 차단한다.

본 논문의 Ⅱ장에서는 관련 연구 및 배경지식을, Ⅲ장에서는 Earlybird 와 Snort를 이용한 웜 차단 방법에 대해 제안하고, 제안된 웜 차단 방법을 Deterlab 환경에서 실험을 통해 그 유효성을 증명하며, Ⅳ장에서 결론을 맺는다.

Ⅱ. 관련연구 및 배경지식

1. Deterlab 소개[3-5]

사이버 공격의 진화를 효과적으로 예측, 탐지 및 대응하기 위해 시작된 사이버 보안 테스트베드 프로젝트인 DETER 프로젝트는 2004년부터 시작되어, 사이버 보안 연구, 테스트 및 평가를 중심으로 운영하고 있다. 이 시설은 남가주 대학교(USC University of Southern California)의 ISI (Information Sciences Institute) 및 캘리포니아 버클리 대학에 위치하고 있으며, 약 400여대의 컴퓨터와 10개의 NetFPGA 등 각종 실험에 필요한 하드웨어 장비 및 소프트웨어로 구성되어 있다. 미국 정부의 지원과 여러 후원사들이 기부하여 만들어진 Deterlab은 공개된 테스트베드이다.

2. Earlybird 소개[6,7]

온라인 침입탐지시스템인 Earlybird system은 크게 Sensor(감지)와 Aggregator(수집) 두 부분으로 구성되어 있다. Sensor는 트래픽 모니터링을 통해 의심되는 시그너처를 탐지하는 역할을 하고, Aggregator는 Sensor로부터 업데이트된 시그너처를 모아서, 네트워크 레벨 또는 호스트 레벨의 차단 서비스에 이용된다.

Earlybird에서는 다양한 출발지와 목적지사이에서 발생하는 트래픽에 일반적인 문자열이 존재하고, 이 특별한 트래픽 패턴이 웜을 감지하기에 충분할 것으로 가정한다 (참고로, CodeRed 웜의 경우 “N” 또는 “X”문자를 포함). 또한 시간의 경과에 따라 랜덤한 출발지, 목적지 주소를 반영하는 웜의 경우, 주소 확산 형태가 일반적인 네트워크 트래픽과 다르게 보이는 점을 이용한다.

Earlybird는 위와 같은 가정을 바탕으로 Content Sifting이라 불리는 과정을 통해, 입력된 패킷의 분석을 통해 시그너처 자체 (문자열, 고유한 데이터)를 생성하는 것이 아니라, 인덱스로 사용되는 해쉬값을 계산하여, 그 값이 같은 횟수를 파악하고, 미리 설정한 임계값을 초과하면 경고를 발생하게 된다. 임계값은 크게 content prevalence 임계값과 address dispersion 임계값으로 구성된다. 일반적인 패킷이 동일한 시그너처를 생성하는 경우는 1번 또는 2번이다. 따라서 동일한 시그너처가 3번 이상 나올 경우, 웜의 가능성이 높기 때문에, 본 실험에서는 content prevalence 임계값을 3으로 사용하고 있다. address dispersion 임계값은 서로 다른 출발지와 목적지 주소사이에 전송되는 패킷에서 같은 시그너처 (문자열 또는 content)가 발견되는 주소의 수를 말한다. 출발지 및 목적지 임계값을 30 이상으로 할 경우, 가장 낮은 탐지율을 보였다. 반대로 2 이상으로 할 경우, 가장 많은 시그너처 탐지가 가능하지만, 거짓 긍정 (false positive)일 가능성이 높다. 일반적으로 address dispersion 임계값은 네트워크 진입 부분, 출발지 및 목적지에 가까운 부분 등

패킷을 추출하는 네트워크 위치에 의존한다. 본 실험에서는 address dispersion 임계값을 10으로 사용하고 있다.

3. Snort 소개[8]

침입탐지시스템(Intrusion Detection System)은 네트워크를 통해 외부에서 들어오는 침입을 감지하는 시스템으로써, 정보시스템의 보안을 위협하는 침입행위가 발생할 경우 이를 탐지하고 적극적으로 대응하기 위한 시스템이다. 오픈 소스 침입탐지시스템인 Snort는 자체적으로 패킷 로깅과 실시간 트래픽 분석기능을 갖추고 있다.

그림 1에서 볼 수 있듯이, Snort는 크게 Libpcap 라이브러리를 이용하여 패킷을 캡처하는 부분, 링크 계층에서 패킷을 분석 및 가공 (패킷 데이터를 구조화 및 정규화)하여 전처리기와 탐지 엔진에게 제공하는 패킷 디코드엔진 부분, 디코드로부터 받은 패킷을 내용 검사, 경고 전송 및 수정 작업을 수행 후 탐지 엔진에게 보내는 전처리기 부분, 기존에 등록된 규칙의 항목과 비교하는 탐지 엔진 부분, 그리고 탐지 엔진 또는 전처리기로부터 생성된 경고를 출력하는 부분으로 나눌 수 있다.

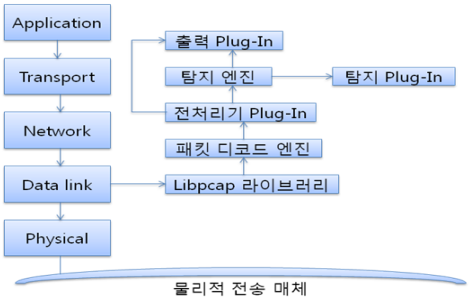

그림 1. Snort 처리과정
Fig. 1. Snort Process

한편, Snort의 규칙에는 입력된 패킷이 규칙을 만족할 때 적용할 행위, 프로토콜의 종류, 출발지와 목적지의 주소 및 포트번호, 침입 발견 시 전송할 메시지의 내용, content(즉, 매칭 검사를 수행할 문자열), 해당 규칙의 참조 내용 등을 포함하고 있다.

III. 제안하는 웜 차단 방법

이번 장에서는 본 과제의 목표인 웜 확산 차단 실험을 위해, Deterlab환경에서 CoderedII 웜 트래픽을 발생시키고, Earlybird 도구를 이용하여 웜 시그너처를 생성한 다음, 이를 Snort 에 적용하여 웜 확산을 차단하는 방법을 설명한다.

1. Earlybird를 이용한 웜 탐지

그림 2는 웜 탐지 실험의 전체적인 구성을 보여준다.

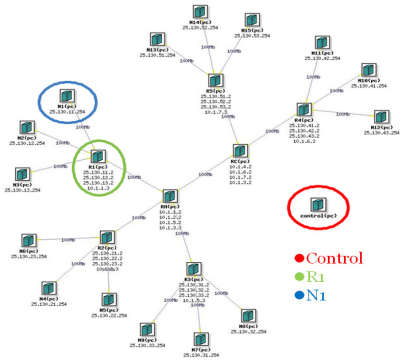

그림 2. 실험 구성도
Fig. 2. Experiment Configuration

총 18대의 노드(컴퓨터 및 라우터 역할)로 구성하여 진행하였고, 실험에 사용된 노드는 3Ghz Intel Xeon 1U server로서 Ubuntu 12.04 64bit 버전이 이용되었다[9]. 또한, 각 노드에는 두 개의 Intel Gigabit Ethernet card 가 장착되어 실험용 네트워크와 관리용 네트워크로 구분한다.

한편, 전체 노드를 실제 환경처럼 다양한 네트워크 대역으로 구성하기 위해서, 그림 2에서 Control 노드는 실험환경의 네트워크 구성을 가상의 다양한 IP대역(그림 3에서 서로 다른 15가지 네트워크 대역으로 구성됨)으로 맵핑하는 역할을 수행한다. N1 노드의 경우, CodeRedII 웜을 전파하는 최초 근원지 역할을 한다. 실제 알려진 CodeRedII 웜의 공격은 상용 웹서버를 타겟으로 TCP 80번 포트를 통해, Buffer Overflow 공격을 진행한다.

```
tb-set-ip-link $N1 $linkR11 25.130.11.254        rt.setForNode('range', 'N1', '69.130.1.1/24')
tb-set-ip-link $N2 $linkR12 25.130.12.254        rt.setForNode('range', 'N2', '69.130.2.1/24')
tb-set-ip-link $N3 $linkR13 25.130.13.254        rt.setForNode('range', 'N3', '69.60.3.1/24')
tb-set-ip-link $N4 $linkR21 25.130.21.254        rt.setForNode('range', 'N4', '69.60.4.1/24')
tb-set-ip-link $N5 $linkR22 25.130.22.254        rt.setForNode('range', 'N5', '69.60.5.1/24')
tb-set-ip-link $N6 $linkR23 25.130.23.254        rt.setForNode('range', 'N6', '89.130.6.1/24')
tb-set-ip-link $N7 $linkR31 25.130.31.254        rt.setForNode('range', 'N7', '89.130.7.1/24')
tb-set-ip-link $N8 $linkR32 25.130.32.254        rt.setForNode('range', 'N8', '89.251.8.1/24')
tb-set-ip-link $N9 $linkR33 25.130.33.254        rt.setForNode('range', 'N9', '89.251.9.1/24')
tb-set-ip-link $N10 $linkR41 25.130.41.254       rt.setForNode('range', 'N10', '89.251.10.1/24')
tb-set-ip-link $N11 $linkR42 25.130.42.254       rt.setForNode('range', 'N11', '100.60.11.1/24')
tb-set-ip-link $N12 $linkR43 25.130.43.254       rt.setForNode('range', 'N12', '100.60.12.1/24')
tb-set-ip-link $N13 $linkR51 25.130.51.254       rt.setForNode('range', 'N13', '100.251.13.1/24')
tb-set-ip-link $N14 $linkR52 25.130.52.254       rt.setForNode('range', 'N14', '100.251.14.1/24')
tb-set-ip-link $N15 $linkR53 25.130.53.254       rt.setForNode('range', 'N15', '100.251.15.1/24')
```

그림 3. 실험에 사용된 IP 구성
Fig. 3. IP Configuration

한편, 네트워크 내에서 패킷을 수집하는 장소가 바로 R1 노드(최초 웜 공격 근원지에 위치한 라우터)이고, 그림 4는 여기서 수집된 패킷을 Earlybird를 이용하여 탐지한 결과를 나타낸다. 이는 시간의 경과에 따른 의심되는 패킷을 발견하고 경고를 출력하는 모습이다.

```
Alarm time-1342500533 sig-7548017595200307206
Alarm time-1342500535 sig-7548017595200307206
Alarm time-1342500536 sig-7548017595200307206
Alarm time-1342500546 sig-7548017595200307206
Alarm time-1342500548 sig-7548017595200307206
Alarm time-1342500549 sig-7548017595200307206
Alarm time-1342500552 sig-7548017595200307206
```

그림 4. Earlybird의 경고 출력 화면
Fig. 4. Alarm Display in Earlybird

2. 패킷분석 및 Snort 규칙 작성

그림 5는 R1노드에서 탐지된 패킷을 Wireshark를 이용하여 분석하는 과정을 나타낸다.

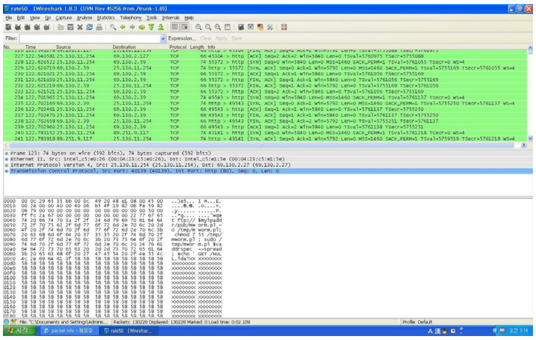

그림 5. 수집된 패킷 분석
Fig. 5. Packet Analysis using Wireshark

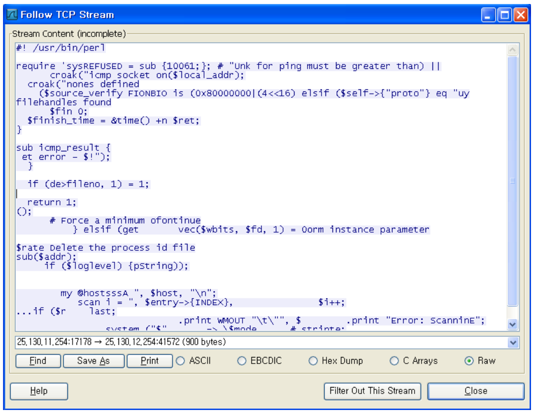

그림 6. 내용 확인
Fig. 6. Verification of Content

그림 6의 상단 그림은 Wireshark 의 Follow TCP Stream 기능을 이용하여서 Client/Server 간에 주고받는 문자열을 보여준다. 이러한 문자열 중에서 의심되는 부분(예: 웜 확산에 이용되는 파일 전송명령어 및 웜 파일 실행 명령어)을 확인함으로써, 웜이라고 추정할 수 있다.

이렇게 추정된 부분을 바탕으로, 그림 7과 같이 Snort 규칙을 작성할 수 있다.

첫 번째 규칙은 mworm.pl 이라는 파일을 다운받기 위한 명령어로써, 해당 명령어를 포함하는 패킷을 차단할 필요가 있다.

두 번째 규칙은 다운받은 mworm.pl 파일의 권한을 변경(기존에는 다운받은 유저만이 접근할 수 있었으나, 모든 사용자가 해당 파일을 실행하거나 읽을 수 있도록 변경)하는 명령어로써 규칙에 적용하였다.

세 번째 규칙은 mworm.pl 파일을 실행하는 명령어를 차단하기 위해서 적용한 부분이다.

끝으로, 마지막 규칙은 CodeRedⅡ의 고유 문자열 시그너처를 규칙에 적용한 부분이다.

```
alert tcp any any -> any any
(msg:"wget ftp CodeRed2 by
earlybird"; content:"wget
ftp://$myipaddr/pub/mworm.
pl -O /tmp/mworm.pl";
sid:1004; rev:7;)

alert tcp any any -> any any
(msg:"chmod CodeRed2 by
earlybird"; content:"chmod
755 /tmp/mworm.pl";
sid:1005; rev:7;)

alert tcp any any -> any any
(msg:"spread CodeRed2 by
earlybird"; content:"sudo
/tmp/mworm.pl $vaddrspec
--spread"; sid:1006; rev:7;)

alert tcp any any -> any any
(msg:"streaming XXXX
CodeRed2 by earlybird";
content:"echo 'GET /NULL.ida?
XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX
XXXX'"; sid:1007; rev:7;)
```

그림 7. Snort 규칙 작성
Fig. 7. Writeup of Snort Rule

3. Snort를 이용한 웜 차단

Snort는 정상적인 탐지가 이루어지면 alert.ids 파일과 snort.log 파일을 생성한다. 그 중 alert.ids 파일에는 공격에 대한 이벤트 명, 공격의 방향(출발지 주소로부터 목적지 주소로 진행하는 방향), 프로토콜 정보가 담겨있다. 그림 8은 탐지된 alert 메시지의 정보를 보여준다. 여기서 이벤트명은 CodeRedⅡ by earlybird이고, 공격 방향은 25.130.12.254:41506 -> 25.130.11.254:21이며, 이용된 프로토콜은 TCP이다.

```
C:\Snort\log\ddos log\alert.ids
[**] [1:230:7] CodeRed2 by earlybird [**]
[Priority: 0]
07/19-16:38:35.435013 25.130.12.254:41506 -> 25.130.11.254:21
TCP TTL:63 TOS:0x0 ID:46279 IpLen:20 DgmLen:65 DF
***AP*** Seq: 0x450DDDA3 Ack: 0x45ACAF5A Win: 0x5B4 TcpLen: 32
TCP Options (3) => NOP NOP TS: 107505969 107505930

[**] [1:230:7] CodeRed2 by earlybird [**]
[Priority: 0]
07/19-16:38:39.558056 25.130.12.254:41508 -> 25.130.11.254:21
TCP TTL:63 TOS:0x0 ID:45098 IpLen:20 DgmLen:65 DF
***AP*** Seq: 0x4580C976 Ack: 0x45E1E04A Win: 0x5B4 TcpLen: 32
TCP Options (3) => NOP NOP TS: 107510092 107510052

[**] [1:230:7] CodeRed2 by earlybird [**]
[Priority: 0]
07/19-16:38:43.660368 25.130.12.254:41510 -> 25.130.11.254:21
TCP TTL:63 TOS:0x0 ID:21043 IpLen:20 DgmLen:65 DF
***AP*** Seq: 0x466BDB76 Ack: 0x45A15128 Win: 0x5B4 TcpLen: 32
TCP Options (3) => NOP NOP TS: 107514195 107514155
```

그림 8. 탐지된 alert 메시지
Fig. 8. Detected Alert Message

한편, snort.log 파일에는 해당 공격에 대한 패킷 정보가 담겨 있다. 그림 9는 탐지된 로그 정보를 보여준다. 수집된 패킷들 중에서 Snort 규칙에 적용되어 걸러진 패킷 정보를 보여준다.

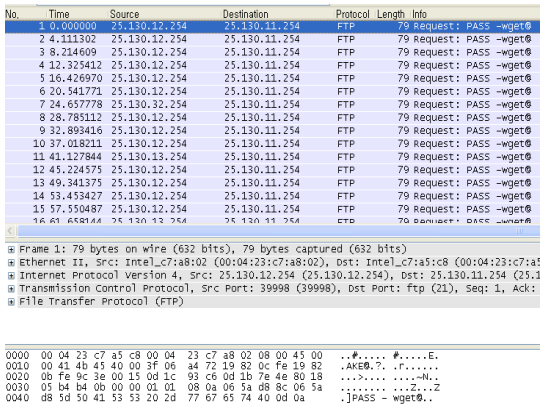

그림 9. Snort Log 파일 내용
Fig. 9. Snort Log File

Ⅳ. 결론

본 논문에서는 가상 시뮬레이션 테스트베드인 Deterlab 환경에서, CoderedⅡ 웜 트래픽을 발생시켰다. 이를 Earlybird를 이용하여 의심스러운 부분을 식별한 후, Wireshark를 통해 분석하여 Snort 규칙을 작성하였다. 다음으로, CoderedⅡ 웜 트래픽에 Snort(이전에 작성된 규칙이 적용됨)를 적용함으로써, 생성된 로그파일의 확인을 통해, 정상적으로 웜 탐지가 이루어짐을 확인할 수 있었다.

하지만, 앞에서 소개된 Earlybird의 웜 탐지 방법은 특정한 경우, 즉 랜덤한 주소로 확산되는 웜 또는 바이러스 탐지에만 적용가능하다. 그러나, DDoS, 스팸메일과 같이 특정 타겟을 목표로 하는 공격과 계속해서 변형되는 다형성 바이러스에는 적용되기 어려운 단점이 있으므로 이 부분에 대한 추가 연구가 필요하다. 또한, Earlybird를 이용한 탐지 시 관리자가 address dispersion 임계값, content prevalence 임계값을 적절하게 정의를 해주어야 오탐지율을 줄일 수 있다. 따라서, 앞으로 다양한 상황에 맞는 적절한 임계값을 구하는 방법 또한 연구되어야 한다.

참 고 문 헌

[1] Songjie Wei, Jelena Mirkovic, Martin Swany “Distributed Worm Simulation with a Realistic Internet Model” Computer and Information Sciences University of Delaware Newark.

[2] Cliff Changchun Zou, Weibo Gong, Don Towsley “Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense” Dept. Electrical and Computer Engineering Univ. Massachusetts Amherst.

[3] Songjie Wei, Calvin Ko, Jelena Mirkovic, Alefiya Hussain “Tools for Worm Experimentation on the DETER Testbed”.

[4] http://www.deter-project.org/.

[5] http://isi.deterlab.net/index.php3.

[6] Sumeet Singh, Cristian Estan, George Varghese, Stefan Savage “Automated Worm Fingerprinting” Department of Computer Science and Engineering, University of California, San Diego.

[7] Sumeet Singh, Cristian Estan, George Varghese, Stefan Savage "The EarlyBird System for Real-time Detection of Unknown Worms" University of California, San Diego.
[8] http://www.snort.org.
[9] https://trac.deterlab.net/wiki/NodeTypes.

※ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2012002139)

저자 소개

이 형 윤(준회원)

• 2013년 : 홍익대학교 컴퓨터정보통신공학과(학사)
<주관심분야 : 정보보호>

황 성 운(정회원)

• 1993년 : 서울대학교 수학과 (학사)
• 1998년 : 포항공과대학교 정보통신학과 (석사)
• 2004년 : 한국과학기술원 전자전산학과 (박사)
• 2008년 ~ 현재 : 홍익대학교 컴퓨터정보통신공학과 교수
<주관심분야 : 정보보호, 프로그래머블 로봇>

안 병 구(종신회원)

• 1988년 : 경북대학교 전자공학과 (학사)
• 1996년 : (미)Polytechnic University, Dept. of Computer and Electrical Eng., USA (석사)
• 2002년 : (미)New Jersey Institute of Technology (NJIT), Dept. of Computer and Electrical Eng., USA.(박사)
• 1989년 ~ 1994년 : 포항산업과학기술연구원(RIST), 선임연구원
• 2003년 ~ 현재 : 홍익대학교 컴퓨터정보통신공학과 교수
• 2012년 ~ 현재 : 대한전자공학회 컴퓨터소사이어티 회장
<주관심분야 : Wireless Networks, Ad-hoc & Sensor Networks, Multicast Routing, QoS Routing, Cross-Layer Technology, Cooperative Communication, Network Coding, Bioinformatics, LED Communication>