情報保護學會論文誌
第18卷 第2號, 2008. 4

단순 추정량을 이용한 악성코드의 탐지척도 선정*

문 길 종[1†], 김 용 민[2‡]

[1]전남대학교 정보보호협동과정, [2]전남대학교 전자상거래전공

Selection of Detection Measures for Malicious Codes using Naive Estimator*

Gil-Jong Mun[1†], Yong-Min Kim[2‡]

[1]Interdisciplinary Program of Information Security, Chonnam National University

[2]Dept. of Electronic Commerce, Chonnam National University

요 약

네트워크 내의 다양한 악성코드는 변종들이 빠르게 생성되고 그 행위는 점차 지능화되어 피해도 커지고 있다. 본 논문에서는 효과적인 악성코드 탐지를 위해 탐지규칙 생성에 효과적인 척도선정 방법을 제안한다. 실험에 헤더 정보만을 활용함으로써 페이로드 데이터를 검사하는 과부하를 최소화 하였고, 패킷의 단순한 정보가 아닌 네트워크 연결정보인 다양한 척도를 사용하여 악성코드의 특징 파악을 용이하게 한다. 실험에 사용된 80개의 연결정보 중 유용한 탐지척도를 선정하기 위해 히스토그램 방법을 이용해 확률 분포를 구하고, 단순 추정량에 적용한 후, 상대 복잡도를 이용한다. 단순 추정량 방법은 기존 방법인 히스토그램 방법의 단점인 임의로 나눈 경계 부근의 값에 대한 오분류를 해결하고, 악성코드 탐지에 유용한 척도의 선택을 유도한다. 선정된 척도를 바탕으로 탐지규칙을 생성하고, 탐지실험을 하여, 그 결과를 기존 방법과 비교 평가함으로써 제안하는 기법이 유용함을 보인다.

ABSTRACT

The various mutations of the malicious codes are fast generated on the network. Also the behaviors of them become intelligent and the damage becomes larger step by step. In this paper, we suggest the method to select the useful measures for the detection of the codes. The method has the advantage of shortening the detection time by using header data without payloads and uses connection data that are composed of TCP/IP packets, and much information of each connection makes use of the measures. A naive estimator is applied to the probability distribution that are calculated by the histogram estimator to select the specific measures among 80 measures for the useful detection. The useful measures are then selected by using relative entropy. This method solves the problem that is to misclassify the measure values. We present the usefulness of the proposed method through the result of the detection experiment using the detection patterns based on the selected measures.

Keywords : *naive estimator, malicious codes, detection pattern, detection measure*

접수일: 2007년 8월 3일; 채택일: 2008년 1월 3일

* 이 논문 또는 저서는 2006년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2006-331-D00560)

† 주저자, alcor@lsrc.jnu.ac.kr

‡ 교신저자, ymkim@chonnam.ac.kr

Ⅰ. 서 론

네트워크는 현대인의 삶에 편의를 제공하고 고도의 정보화 사회를 실현하기 위해 필요한 핵심 기술이다. 하지만 그 개방성과 방대함으로 많은 역효과가 발생하고 있다. 특히 네트워크 및 시스템의 취약점을 악용해서 침투하는 웜(worm)들은 스스로 복제 및 전파되어 사용자 개인정보 침해와 주변 네트워크 장악 등의 피해를 준다. 악성 봇(bot)은 이런 목적을 위해 등장한 가장 대표적인 공격기법이다. 공격자는 수많은 시스템들에 침투하여 악성 봇을 설치하고 이를 이용해 피해 시스템들을 공격자가 관리할 수 있는 네트워크로 연결하여 명령을 따르게 한다. 이를 봇넷(botnet)이라 하며 피해 시스템들은 주변의 시스템들을 계속해서 감염시킨다. 일반적으로 악성코드 탐지 시스템은 공격 시그니처에 의존하고 있으며, 모든 변종 악성코드들에 대해 각각의 시그니처를 보유하기 위해 주기적으로 전문가에 의해 업데이트를 수행해야하는 어려움이 존재하며, 이때 탐지 영역이 되는 페이로드는 고성능 네트워크 환경에서 모든 패킷을 분석해야 하므로 많은 부하가 발생한다. 이러한 이유로 자동으로 탐지규칙을 생성하고 패킷 헤더 정보만을 이용한 빠른 탐지 방법이 요구된다.

네트워크에서 패킷들은 연결정보보다 단편적인 정보만을 가지고 있어 정상과 악성코드를 분류 및 분석하기 힘들다. 본 논문에서는 각 연결정보 중, *Complete_connection, SYN_counts, FIN_counts, Port* 등의 80가지의 정보를 추출하여 각 악성코드에 대한 각 척도의 확률분포를 히스토그램 방법을 통해 구한다. 확률 분포를 바탕으로 단순 추정량 방법을 적용하여 임의로 나눈 경계값에 의해 오분류된 확률 분포를 보완하고 탐지에 유용한 척도를 상대 복잡도를 이용해서 선정하는 방법을 제안한다. 선정된 척도를 이용하여 의사결정나무 알고리즘 중 하나인 C4.5에 의해 탐지규칙을 생성하고 변환하여 탐지한다. 제안된 방법의 정확성을 검증하기 위하여 선정된 탐지척도와 개수를 제시했고, 악성코드의 분류 및 탐지, 오탐 결과를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서 악성코드 탐지에 대한 연구, 네트워크 프로토콜 기반 공격분석 연구 및 본 논문의 선행연구에 대해 기술한다. 3장에서는 본 논문에서 제안하는 탐지 시스템, 탐지척도, 척도선정 및 탐지 규칙 생성에 관한 방법을 설명한다. 4장은 실험 및 결과분석으로써 실험에 사용된 데이터, 척도선정 결과, 분류율과 오탐율을 이용한 실험결과를 설명하고 5장에서 결론을 맺는다.

Ⅱ. 관련연구

2.1 악성코드 탐지

Autograph는 CMU의 연구 프로젝트로 웜으로 추측되는 패킷들을 추출하여 자동으로 웜에 대한 시그니처를 생성한다[1]. 이는 비정상행위 탐지 기법과 비슷한 비감독 학습으로 생성된 시그니처를 이용하여 빠르게 오용행위 탐지를 수행하게 된다. 생성된 시그니처는 다른 침입탐지 또는 침입방지 시스템에 적용될 수 있는 규칙으로 규격화 될 수 있다. 이 방법은 웜이라고 의심되는 패킷들을 수집하고 페이로드 데이터 영역에 대한 검사를 수행하여 COPP(COntent-based Payload Partitioning)를 작성하게 된다. 이를 토대로 유사성 알고리즘인 Rabin's Fingerprint를 이용하여 공통 규칙을 찾아내는데 실제 구현이 매우 간단하고 동작이 빠르기 때문에 많은 곳에서 활용되고 있다. Autograph는 비감독 학습이 가능하고 분산환경에 적용이 가능하다는 장점을 지니지만, 비스캐닝 웜은 탐지가 불가능하고 패킷의 데이터 영역에 의존하므로 탐지 부하가 다소 가중된다. 그리고 오탐율을 최소로 하기 위하여 탐지대상에서 제외되어야 하는 특징리스트의 작성에 전문가의 도움이 요구되어 진다.

DEWP(Detecting Early Worm Propagation through Packet Matching)는 ISI에서 제안한 알고리즘으로 네트워크상에 유입되는 트래픽들을 특정 규칙들로 분류하여 웜을 탐지한다[2]. 일반적으로 정상적인 트래픽에 비해서 웜은 특정 네트워크 서비스 포트로 유입되는 트래픽의 양이 많게 된다. 웜의 이러한 성질을 이용해서 특정 포트가 다른 포트에 비하여 혹은 이전에 비하여 갑자기 많은 트래픽들을 받게 되면 이를 웜으로 의심하고 탐지하는 것이다. 이는 네트워크 패킷의 데이터 영역에 대한 검사를 수행하지 않아도 되며, 단지 특정 포트로 전달된 패킷들과 그 포트에서 다시 외부로 전송된 패킷들의 양적인 정보만을 필요로 한다.

매 시간동안 특정 포트로 전달된 패킷의 양과, 그 포트에서 나간 양을 비교하여 EWMA(Exponential Weighted Moving Average) 알고리즘을 이용하여 변화량을 추적하게 되고 변화가 발생하면 이를 웜 공격으

〔표 1〕 악성코드 탐지 시스템의 비교

연구	장점 및 단점
Auto-graph	- 실제 구현이 매우 간단하고 동작이 빠름 - 비감독 학습이 가능 - 분산환경에 적용이 가능
	- 스캐닝하지 않는 웜은 탐지가 불가능 - 패킷의 데이터 영역에 의존하므로 탐지 부하 발생 - 오탐율을 낮추기 위해 탐지대상에서 제외되어야 하는 특징리스트의 작성에 전문가의 도움이 요구
DEWP	- 쉽고 간편하게 웜을 찾아 낼 수 있음 - TCP와 UDP 웜 모두에 적용이 가능
	- 갑자기 외부 접속자가 많아지는 사이트의 경우 웜과 일반 사용자의 빈도가 비슷해질 수 있으므로 이에 대한 오탐이 증가
제안 방법	- 패킷 헤더 정보만을 이용한 탐지 - 전문가의 도움 없이 탐지규칙을 자동으로 생성 - 오탐 가능성을 최소화하기 위해 각 세션의 통계적 정보를 이용
	- UDP 악성코드에 대한 실험이 없었음 - 다양한 분류의 악성코드가 요구됨

로 간주하게 된다. 이러한 방법은 쉽고 간편하게 웜을 찾아 낼 수 있고 TCP와 UDP 웜 모두에 적용이 가능하지만, 갑자기 외부 접속자가 많아지는 사이트의 경우 웜과 일반 사용자의 빈도가 비슷해지는 경우가 발생하기 때문에 이에 대한 오탐이 증가할 것이라는 단점이 있다. [표 1]은 Autograph, DEWP와 본 논문에서 제안한 방법의 장·단점을 비교한다.

2.2. 네트워크 프로토콜 기반 공격분석 연구

오하이오 대학에서는 IP와 TCP의 패킷분석을 통해 의심스러운 패킷들의 의미와 패킷이 갖는 영향을 평가하였다. IP 헤더의 패킷 크기, IP 체크섬, TTL 필드의 값, IP 주소, IP 옵션, 겹치는 데이터와 TCP 헤더의 패킷 크기, TCP 체크섬, 포트 번호, TCP 플래그, 예약 비트들에 대해 분석하였다. 이와 같은 필드들을 대상으로 에러 비율과 패킷이 차지하는 비율을 통계적으로 분석하여 네트워크의 이상을 분석하였다[3]. 그러나 이 연구에서는 실제 공격에 의해 발생된 패킷에 대한 분석은 없었으며 자체 망에서의 패킷의 이상 유무에 대한 특성만을 분석하였다.

플로리다 대학에서는 네트워크 프로토콜인 Ethernet, IP, TCP, UDP, ICMP 패킷 헤더값의 비정상적인 분포를 분석하기 위해 패킷 이상점수를 계산하는 PHAD (Packet Header Anomaly Detection)[4]와 베이즈 규칙을 이용하는 ALAD (an Application Layer Anomaly Detector)[5] 알고리즘을 사용했다. 이 연구에서는 선택한 공격특징이 TTL 필드 등으로써 공격기법과 관련성이 적었으며 탐지된 결과로 공격현상을 설명하기에는 미흡하였다. 또한 패킷의 데이터(payload)에 들어있는 애플리케이션 계층에서의 공격들은 탐지하지 못하는 단점을 가지고 있었다. 그리고 PHAD와 ALAD 알고리즘을 같이 사용하는 경우에 완벽하지 못한 결과를 나타냈다.

〔표 2〕 네트워크 프로토콜 기반 탐지 비교

기관	특징
오하이오	- IP와 TCP의 패킷분석을 통해 비정상적인 패킷들의 의미와 패킷이 지닌 영향을 평가 - 에러 비율과 패킷이 차지하는 비율을 통계적으로 분석하여 네트워크의 이상을 분석
	- 실제 공격에 의해 발생된 패킷에 대한 분석은 없음 - 패킷의 이상 유무에 대한 특성만을 분석
플로리다	- Ethernet, IP, TCP, UDP, ICMP 패킷 헤더값의 비정상적인 분포를 분석 - PHAD와 ALAD 알고리즘 사용
	- 선택한 공격특징이 공격기법과 관련성이 적었음 - 탐지된 결과로 공격현상을 설명하기에는 미흡함
U.C. Davis	- 호스트와 라우팅 기반으로 위조 패킷의 탐지 - 소스 IP 주소를 기반으로 TTL 값에 대해서 평가 - 라우팅과 비라우팅 방법 사용
	- 패킷위조 공격유형에 대해서만 분석 - 연구 이외의 다른 유형공격이나, 다른 척도들에서 연구특성을 나타내는 공격의 경우에는 탐지되지 않음
제안방법	- 패킷의 이상 유무 뿐 아니라 특징을 설명 - 공격에 대한 관련성 있는 척도 및 특징을 찾기 위해 척도 선정 - 생성된 규칙으로 공격현상을 설명할 수 있음 - 패킷 위조 뿐만 아니라 여러 형식의 공격에 대한 탐지 가능

U.C. Davis 대학에서는 위조된 패킷들을 탐지하기 위해 호스트 기반과 라우팅 기반의 방법을 연구하였다. 이 방법들을 통해 수신한 패킷들이 위조된 근원지 주소를 가지고 있는지 여부를 결정하는데 도움이 되었다. 이 연구에서는 패킷 위조 공격기법들을 분석하였고, 이와 같은 공격들을 라우팅 방법과 비라우팅 방법으로 탐지하였다[6]. 이 연구에서는 근원지 IP 주소를 기반으로 주로 TTL 값에 대해서 평가하였다. 따라서 다른 유형

의 공격이나 다른 척도들에서 그 특성을 나타내는 공격의 경우에는 탐지되지 않는다는 한계를 가진다. [표 2]는 네트워크 프로토콜 기반 공격분석 연구에 대한 특징과 단점을 설명한다.

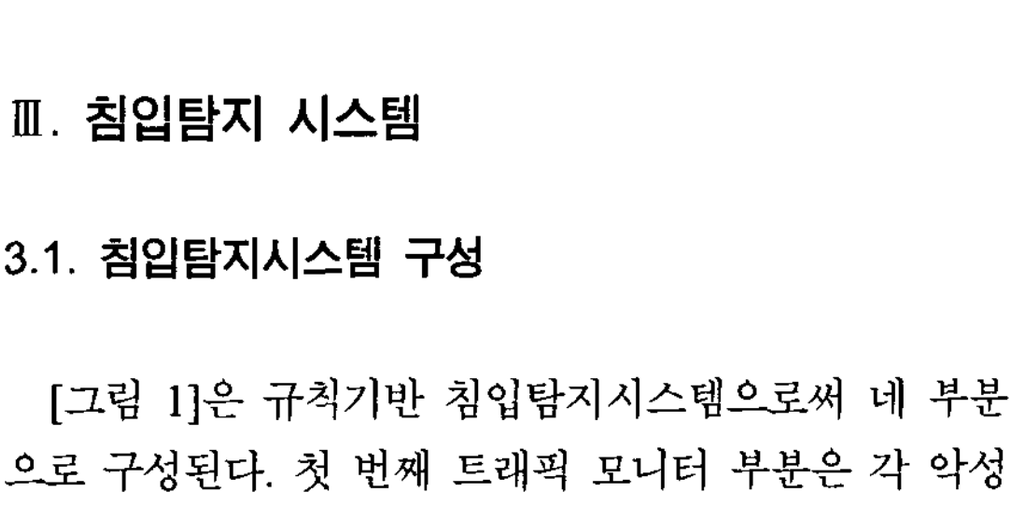
2.3. 네트워크 프로토콜 기반 공격분석 연구

본 논문의 선행연구에서는 KDDCUP 99 데이터 셋을 이용한 연구가 진행되었다. 히스토그램 방법에 의한 척도 선정 및 규칙생성에 대한 연구[7]와 통계적인 기법을 이용한 탐지연구가 선행되었다[8]. 히스토그램 방법을 이용한 연구에서는 침입에 해당하는 각 척도에 대한 확률 분포를 이용한 연구로써 각 침입의 척도에 대한 확률 분포의 정상데이터에 대한 척도 분포 차이를 구해 척도를 선정하고 자동으로 규칙을 생성한다는 장점이 있었지만 확률 분포를 구하기 위해 임의로 경계를 나누었기 때문에 데이터 분포에 대한 오분류가 생긴다는 단점이 존재했다. 통계적 기법은 구해진 확률 분포를 바탕으로 우도비 검증을 이용한 임계값을 설정으로 탐지한다. 이 기법은 임계값 설정에 따라 탐지율과 오탐율을 조정할 수 있었지만 사용자가 임계값을 조정해야한다는 단점이 존재했다.

Ⅲ. 침입탐지 시스템

3.1. 침입탐지시스템 구성

[그림 1]은 규칙기반 침입탐지시스템으로써 네 부분으로 구성된다. 첫 번째 트래픽 모니터 부분은 각 악성코드 별로 패킷들을 세션 단위로 묶고 데이터 처리를 위해 가공한다. 두 번째 부분은 히스토그램(histogram estimator)을 이용해 확률 분포를 구하고 이를 단순 추정량(naive estimator)을 적용한 후, 상대 복잡도(relative entropy)를 이용하여 악성코드 탐지에 유용한 척도를 선정한다. 세 번째 부분은 의사결정나무 알고리즘 중 하나인 C4.5 알고리즘을 이용하여 트리를 생성하고 생성된 트리를 탐지규칙으로 변환한다. 마지막 단계에서 탐지규칙과 세션 데이터를 비교하여 분류 및 탐지한다.



〔그림 1〕 규칙기반 침입탐지 시스템

3.2. 탐지척도

악성코드의 분류 및 탐지를 위해 패킷이 아닌 패킷들의 묶음인 한 개의 연결 단위로써 패킷 그룹화를 수행한다. 각 패킷의 정보가 아닌 한 개의 연결에 대한 통계적인 정보들을 구하고 각각을 그 세션의 데이터로써 활용한다. 하나의 세션에서 뽑아낸 통계적인 척도는 모두 80가지[9]이고 [표 3]과 같이 구분할 수 있다. TCP 세션에서 얻을 수 있는 척도를 통합적, 양방향, 단방향의 세 가지로 분류하고 이산적(categorical) 또는 연속적(continuous)인 값을 갖는 속성으로 나눈다. 세션에서 추출한 척도들은 대부분 두 호스트들(서버-클라이언트)

〔표 3〕 TCP 연결에 대한 80개의 통계적 척도

분류	속성	척도
통합	이산	Complete_connection, SYN_counts, FIN_counts
	연속	Elapsed_time, Packet_counts
양방향 (HB)	이산	HAsyn_count, HAfin_count, HAurg_data, HAurg_bytes
	연속	HApackets, HAreset_sent, HAack_sent, HApureack_sent, HAsack_sent, HAdsack_sent, HAmaxack_sent, HAuniq_sent, HAdata_pkts, HAdata_bytes, HArexmit_pkts, HArexmit_bytes, HAnum_zwnd_probes, HAzwnd_probe_bytes, HAout_order_pkts, HAdata_pkts_push, HAmss_data, HAmaxseg_size, HAminseg_size, HAavgseg_size, HAwin_max, HAwin_min, HAwin_zero_ct, HAavg_win_adv, HAinitialwin_bytes, HAinitialwin_segs, HAttl_length, HAmissed_data, HAtrunc_data, HAtrunc_packet,, HAdata_mixed_time, HAidletime_max, HAthroughput
단방향	연속	HBport

```
N,1,0,8.947825,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,2,0,0
,0,0,0,0,0,0,0,3,0,0,0,0,0,0,0,1440,0,0,0,0,0,0.000000,0.000000,
16384,0,16384,0,0,0,0.000000,0.000000,0,0,0,0,0,0,0,0,0,0,0,0,0,
0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,445,
worm_Sasser.
Y,2,2,0.061286,35,12,23,0,0,11,23,9,2,0,0,0,0,0,0,492,26789,1,
20,492,26789,0,0,0,0,0,0,0,0,0,0,1,4,1,1,1,1,0,0,0,0,1460,1380,4
92,1380,492,569,491.508492,1339.383031,65535,6432,60015,5
840,0,0,0.000000,0.000000,492,2760,1,2,492,26789,0,0,450,25
949,1,20,0.000000,0.030484,9.055000,15.670000,8027.934602,
437114.512287,80,Normal.
```

〔그림 2〕 TCP 세션의 척도값 가공

사이의 연결 정보이다. 양방향 척도에서 접두어 HA는 호스트 A에서 발생한 세션의 정보이고, HB는 호스트 B에서의 세션 정보이다. [표 3]의 양방향 척도 중 접두어 HB로 시작되는 척도는 생략되었다. 추출된 80가지의 척도들은 [그림 2]에서 보는 것과 같이 가공되고 마지막 척도 뒤에는 각 웜과 봇의 명칭이 클래스로써 삽입되며 정상이면 'Normal'이 삽입된다.

3.3. 척도선정

웜과 봇 바이러스 탐지를 위해 80개의 척도에 대한 데이터를 추출하지만, 모든 척도들이 정상과 각 악성코드들을 구분할 수 있는 특징을 가지고 있는 것은 아니다. 본 논문에서는 각각의 웜과 봇을 정상 데이터와 다른 특징을 갖는 척도를 찾아내기 위해 히스토그램 방법으로 확률 분포를 구하고 이를 단순 추정량(naive estimator)[10]에 적용한다. 단순 추정량은 확률 분포를 구할 때 사용되는 히스토그램의 구간의 경계 부근의 확실하지 않는 값이 잘못 분류되는 것을 보완하는 방법이다. 단순 추정량은 각각 나눠진 확률 분포 구간을 정해진 윈도우 크기에 따라 확률 분포값을 반복적으로 포함하여 계산하므로, 경계값에 의해 불분명하게 나눠진 값을 다음 구간값에 반복적으로 포함하여 계산한다. 윈도우 크기는 히스토그램 방법에 의해 나눠진 구간을 반복적으로 포함하는 구간의 개수이다. 식(1)에서 n은 전체 데이터의 개수이고, h는 윈도우 크기에 따른 구간 값이다 ($|x|<1$일 때, $w(x)= 1/2$).

$$f(x)=\frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}w\left(\frac{x-x_i}{h}\right) \quad (1)$$

유용한 척도 선정의 방법으로써 상대 복잡도를 사용한다. 상대 복잡도는 비정상행위 탐지 시 훈련 데이터와 테스트 데이터의 유사함을 판단하기 위해 사용되기도 한다[11]. 각 척도에 대한 상대 복잡도가 작으면 두 데이터에 대한 척도는 비슷한 형태의 데이터로 판단을 할 수 있고, 높으면 높을수록 전혀 다른 데이터로 판단할 수 있다.

식 (2)에서 X는 척도의 벡터(vector) 집합 $X \in \{ x_1, x_2, \ldots , x_N\}$를 나타내는 랜덤변수(random variable)이며, N은 척도의 전체 개수이다. 그리고 각 척도는 서로 독립적이라 가정한다. $p(X)$는 정상에 대한 X의 확률 분포이고, $qj(X)$는 j번째 악성코드에 대한 확률 분포이다($j \in \{1, 2, \ldots , n\}$, n은 악성코드개수). D_{KL}은 정상과 j번째 악성코드에 대한 상대 복잡도이다. $p(X)$와 $qj(X)$의 위치가 바뀌었을 때, D_{KL} 값은 달라지므로 정확한 상대 복잡도의 값을 구할 수 없다. 상대 복잡도 값은 정상과 악성코드의 척도 간의 차이를 계산한 것으로써 거리로 표현한다. Dj는 $D_{KL}(p(X)|qj(X))$의 상대복잡도와 위치가 바뀐 $D_{KL}(qj(X)|p(X))$의 합을 구함으로써, 두 랜덤변수의 위치에 따른 값의 변화를 막을 수 있다.

$$D_{KL}(p(X)\,|\,q_j(X))=\sum_X q_j(X)\ln\left(\frac{q_j(X)}{p(X)}\right)$$
$$D_j=D_{KL}(p(X)\,|\,q_j(X))+D_{KL}(q_j(X)\,|\,p(X)) \quad (2)$$

3.4. 탐지규칙 생성

데이터 마이닝의 분류 기법 가운데 의사결정나무 알고리즘은 분류를 나무 모형과 같은 형태로 자동 구성하여 보여준다. 나무 모형은 최종 노드로 분류되는 개체의 분류 과정을 한 눈에 알 수 있어 목표 개체의 규칙 추출에 용이하다. 더불어 구성되는 척도들의 조건들이 목표 개체에 어떠한 영향을 주는지 쉽게 알 수 있고 목표 개체의 특성을 쉽게 파악할 수 있어 탐지규칙에 적용할 경우 규칙에 해당하는 악성코드와의 연관성을 규명하기에도 적합한 알고리즘이다. 이 가운데서도 C4.5 알고리즘[12]은 엔트로피(entropy)의 감소 값에 기반한 IG (Information Gain)의 값에 의해 척도를 상위 노드로 선정하는 방법을 이용한 나무 모형을 구성하는 알고리즘이다.

학습 데이터를 기반으로 모든 척도 값들의 정보를 추출하고 이어서 탐지에 중요한 척도들만을 선정하여 속성을 정의한다. 이렇게 가공된 학습 데이터를 이용하여, C4.5 의사결정나무 알고리즘은 나무의 노드들을 분리

```
HAmss_data > 1440 :
  | Packet_counts <= 1 :
  |    | HBPort <= 80 :[1642] Normal (183.0)
  |    | HBPort > 80 :
  |    |    | HBPort <= 632 :[1643] worm_Netsky (5.0)
  |    |    | HBPort > 632 :[1644] Normal (39.0)
```

```
Worm_Netsky :
HAmss_data > 1440, Packet_counts <= 1,
 HBPort > 80 : HBPort <= 632
```

〔그림 3〕 의사결정나무에서 변환된 worm_netsky 규칙

해 나가고 나무 모델을 구성한다. 각 노드들의 분리 기준에 의하여 가지를 형성하고, 최종 노드에는 각 가지들에 일치되는 클래스가 정의된다. 최종 노드까지 노드가 확장되어진 경로가 하나의 공격 특징을 갖는 규칙이 된다. [그림 3]은 생성되는 나무모형의 일부를 보여준다. 최대 세그먼트 크기가 요청된 패킷 개수인 척도 *HAmss_data*가 최상위 노드로 결정되어 위치한 후에 수집된 패킷의 총 개수를 나타내는 척도인 *Packet_counts*, 연결 요청 받는 측 포트 번호인 *HBPort* 순으로 노드 분리된 *Worm_Netsky*에 대한 규칙이 변환되는 것을 보여준다. [그림 3]에서와 같이 생성된 탐지규칙 중에서 정상 데이터를 나타내는 *Normal*은 본 논문에서 악성코드로 분류되지 않는 것을 정상 데이터로 판단하기 때문에 탐지규칙에서 제외하고, 실제 탐지에서 또한 사용되지 않는다.

Ⅳ. 실험 및 결과 분석

본 논문에서 실험은 윈도우 크기 1부터 윈도우 크기 10까지, 거리 1부터 10까지 실험을 하였다. 하지만 본 장에서는 기존 방법에서 사용한 윈도우 크기 1과 가장 좋은 결과를 보인 윈도우 크기 4에 대해서 설명한다.

4.1. 탐지규칙 생성

모든 악성코드들을 하나의 단일 의사결정나무로 구성하고 탐지규칙으로 변환한다. 학습 데이터는 악성코드들과 정상행위 데이터로 구성된다. 학습을 통하여 하나의 의사결정나무만이 생성되고 모든 공격규칙들이 표현된다. 생성된 탐지규칙을 이용하여 실험데이터의 공격 여부를 탐지하였다. 학습과 실험에 이용된 정상행위

〔표 4〕 학습에 사용된 데이터의 개수

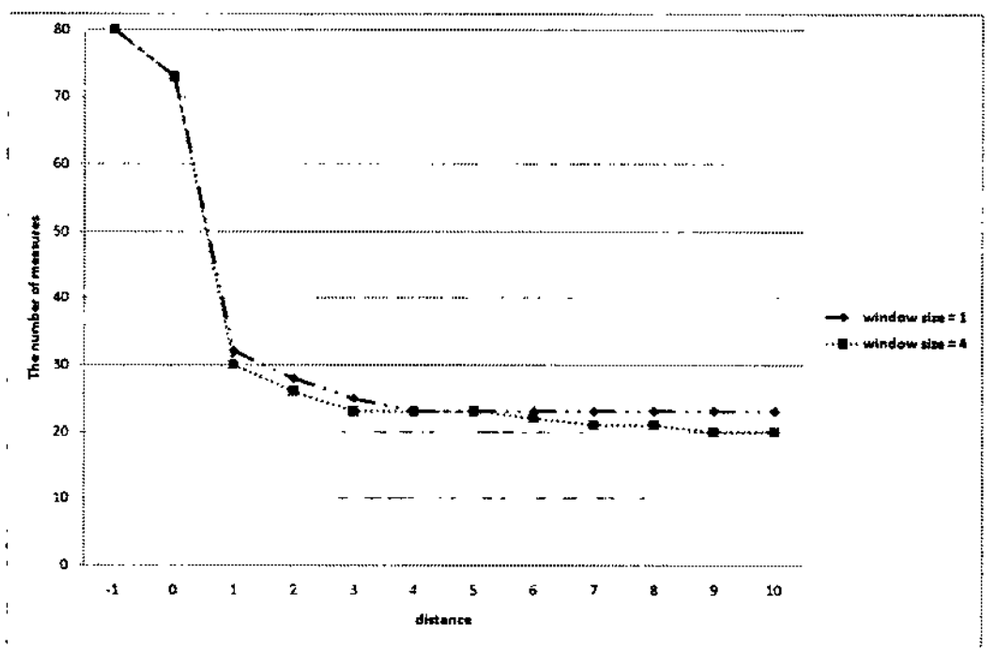
악성 코드	학습 개수	악성 코드	학습 개수
Mimail	5	Gobot	1001
Mydoom	1178	IRCBot	143
Mytob	59	RBot	9534
Netsky	1557	SdBot	1079
Opaserv	27	Wootbot	37
Sasser	5159	poebot	29
Tibick	14	Blaster	729
Welchia	4501	Korgo	341
Bagle	254	Beagle	907
Normal	30725	학습 총 개수	57279

데이터는 자체적으로 구축된 네트워크 환경에서 악성코드 바이너리들을 실행시켜 발생하는 패킷을 직접 수집하여 축적되었으며, 데이터의 2/3는 학습에 사용하고 1/3은 테스트에 사용하였다.

각 웜과 봇에 대해 파생된 변종 웜, 봇들은 본래의 웜과 봇으로 종류(class)를 동일화하였다. 이것은 네트워크 단에서 발생하는 웜과 봇의 파생물들은 패킷의 페이로드 부분이 다르고 네트워크 단에서의 행동과 형태는 유사하다는 가정으로 본 논문에서 제시하고 있는 헤더 정보를 이용하는 것과 맞지 않음으로 이와 같은 방법을 선택했다.

2. 척도선정 결과

네트워크 세션 단위로 뽑아진 80개의 척도가 모두 웜과 봇 탐지를 위해 생성되는 탐지규칙에 유용한 것은 아니다. [그림 4]는 상대 복잡도의 임계값에 대해 기존 방법인 히스토그램의 윈도우 크기 1과 단순 추정량의



〔그림 4〕 거리 변화에 따른 척도의 개수 변화

윈도우 크기를 4로 주었을 때 선택된 척도의 변화를 보여준다. C4.5의 특성상 각 클래스에 척도를 다르게 부여할 수 없으므로 공통된 척도를 사용하게 된다. 거리 -1은 각 척도간의 거리를 계산하지 않기 위해 설정한 것으로 모든 척도가 선택된다. 거리 0은 정상과 악성코드의 데이터에 차이가 없는 것으로 7개의 같은 척도가 제거되어 73개의 선정된 척도를 가지고 결과도 같다. 거리 0에서 1로 증가함으로써 선택된 척도가 각각 32와 30으로 감소됨을 보였다.

3. 테스트 결과

본 논문에서 실험 결과를 분류율, 탐지율, 오탐율을 포함한 표와 분류율과 오탐율을 나타내는 ROC (Receiver Operating Characteristic) 커브[13]로 설명한다. 아래 수식은 논문에서 사용한 분류율, 탐지율, 오탐율에 대한 정의이다. 각 악성코드에 대한 탐지규칙은 그 악성코드와 같은 명칭이 부여된다. 탐지는 정상이 아닌 입력 데이터를 악성코드로 경보한 것을 나타내고, 탐지결과 중에서 악성코드 규칙과 동일한 명칭으로 판명된 것을 분류라고 한다. 그리고 오탐은 악성코드로 경보를 했지만 입력 데이터는 악성코드가 아닌 정상 데이터인 것을 말한다.

$$\text{분류율}(\%)=\frac{\text{악성코드로 분류된 개수}}{\text{악성코드개수}}\times 100$$

$$\text{탐지율}(\%)=\frac{\text{악성코드로 탐지된 개수}}{\text{악성코드개수}}\times 100$$

$$\text{오탐율}(\%)=\frac{\text{정상데이터를 악성코드로 경보한 개수}}{\text{정상코드개수}}\times 100$$

학습 데이터를 바탕으로 모든 악성코드를 하나의 의사결정나무로 구성하고, 이를 바탕으로 생성된 탐지규칙으로 실험한 결과는 [표 5]와 같다. 탐지율은 99%를 선회하고 있어 거의 완벽한 탐지 성능을 보여주고 있고 분류율도 90%에 가까운 높은 비율을 나타냈다. 학습에 사용되어진 많은 악성코드 바이너리들의 행위는 현재 인터넷망의 사전 방어 조치로 인하여 그들의 완전한 공격 절차를 수행할 수 없었으며 감염되기 전 단계 또는 숙주가 된 이후 공격대기 상태의 단계까지의 상태이지만 탐지율에 있어서 높은 성능을 보여주고 있다. 그 이유는 그러한 웜과 봇의 초반 공격행위로 유발되는 특징

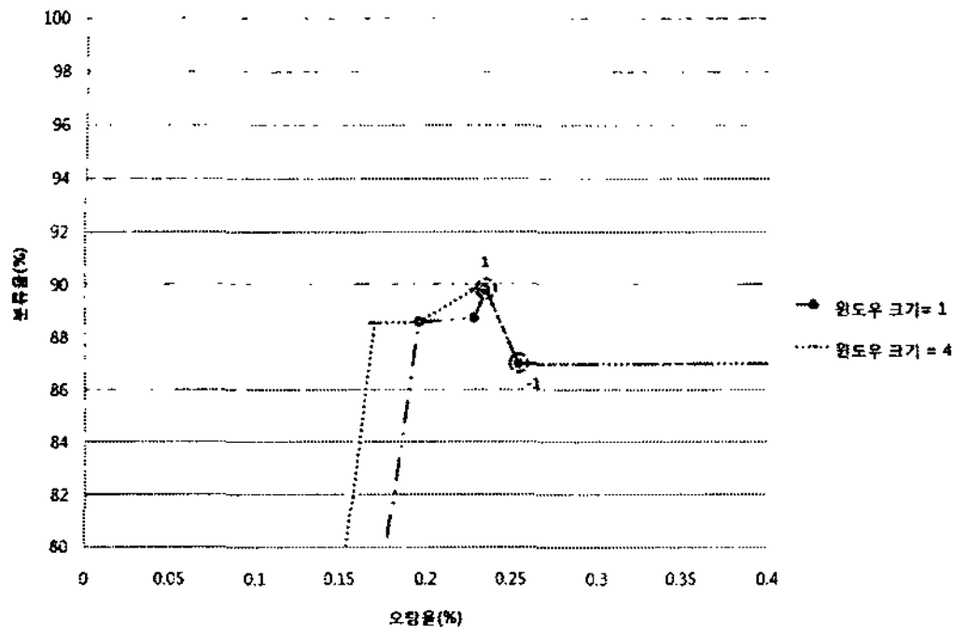
〔표 5〕 거리 1과 윈도우 크기 4일 경우, 탐지와 분류 결과

악성 코드	개수	분류 개수	탐지 개수	분류율 (%)	탐지율 (%)	오탐율 (%)
Normal	15362	-	-	-	-	-
Gobot	500	352	500	70.4	100	-
IRCBot	73	57	73	78.08	100	-
RBot	4768	4460	4733	93.54	99.27	-
SdBot	541	336	535	62.11	98.89	-
Wootbot	19	0	19	0	100	-
poebot	15	5	14	33.33	93.33	-
Bagle	127	69	126	54.33	99.21	-
Blaster	365	365	365	100	100	-
Korgo	170	0	170	0	100	-
Mimail	2	2	2	100	100	-
Mydoom	588	492	584	83.67	99.32	-
Mytob	29	29	29	100	100	-
Netsky	780	768	780	98.46	100	-
Opaserv	13	12	13	92.30	100	-
Sasser	2578	2577	2578	99.96	100	-
Tibick	7	0	7	0	100	-
Welchia	2251	2047	2251	90.94	100	-
Beagle	457	368	450	80.53	98.47	-
결과요약	13283	11939	13229	89.88	99.59	0.23

까지도 정상행위와 구별되는 특징을 나타내고 규칙화가 가능했기 때문으로 분석된다. 또한 봇과 웜을 독립적으로 탐지했을 때보다 여러 악성코드들을 혼용해 탐지했을 때 그 성능이 좋음을 볼 수 있었다. 이를 통해 분석해 볼 때, 의사결정나무는 다양하고 분별력 있는 학습 데이터에 대해 더욱 그 성능이 우수함을 알 수 있다.

[그림 5]는 기존 탐지방법인 히스토그램의 윈도우 크기 1과 단순 추정량의 윈도우 크기를 4로 설정한 후, 거리변화에 따른 분류율과 오탐율에 대한 결과를 나타낸 ROC 커브 그래프이다. 그림에서 '-1'인 부분은 척도 80개가 모두 선택되었을 때, 분류율과 오탐율을 나타낸 것으로써 두 결과에 차이가 없다.

그리고 거리가 1인 부분은 모든 척도를 적용한 것에 비해 높은 분류율과 낮은 오탐율을 보인다. 그래프에서 결과를 나타내는 각 점은 오른쪽에서 왼쪽으로 증가할수록 거리가 증가와 비례하며, 윈도우 크기가 4일 때가 기존 방법인 히스토그램의 윈도우 크기 1일 때보다 분류율은 높고 오탐율은 낮은 이상적인 그래프가 그려짐을 볼 수 있다. [그림 6]은 [그림 5]에서 거리가 1일 경



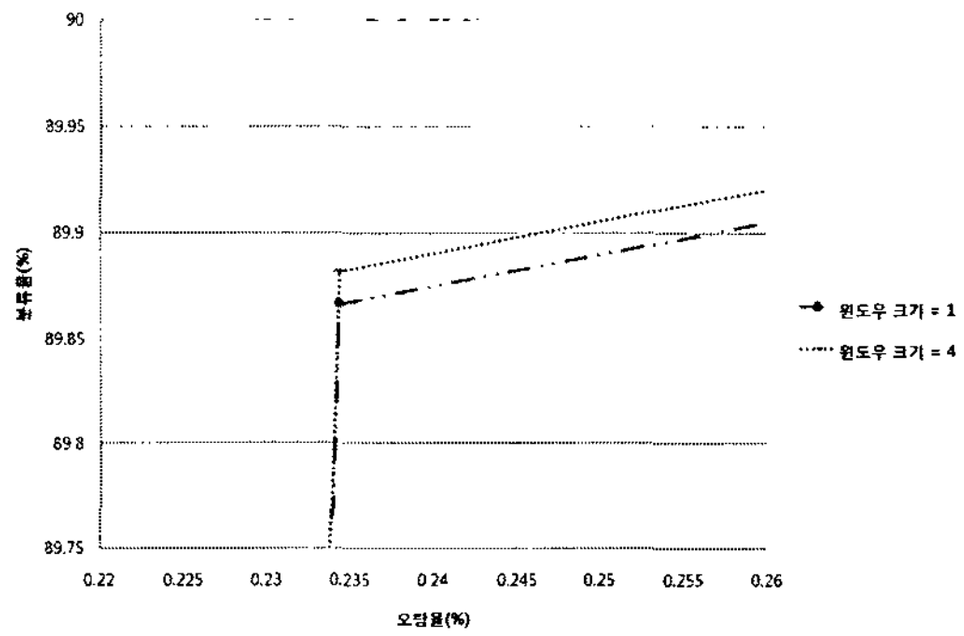
〔그림 5〕 기존 방법인 윈도우 크기 1과 단순 추정량의 윈도우크기 4일 때, 분류율과 오탐율의 변화(그래프의 결과를 나타내는 각 점은 오른쪽에서 왼쪽으로 거리의 증가와 비례함).

〔표 6〕 거리 1일 때, 윈도우 크기 1과 4의 선택된 척도

윈도우 크기	선택된 척도
1	Complete_connection, SYN_counts, FIN_counts, HAreset_sent, HBreset_sent, HAdsack_sent, HAmaxack_sent, HAsyn_count, HBsyn_count, HAfin_count, HBfin_count, HAmss_data, HBmss_data, HAmaxseg_size, HBmaxseg_size, HAminseg_size, HBminseg_size, HAavgseg_size, HBavgseg_size, HAwin_max, HBwin_max, HAwin_min, HBwin_min, HBwin_zero_ct, HBinitialwin_bytes, HBinitialwin_segs, *HAidletime_max*, *HBidletime_max*, HAthroughput, HBthroughput, HBport
4	Complete_connection, SYN_counts, FIN_counts, HAreset_sent, HBreset_sent, HAmaxack_sent, HAsyn_count, HBsyn_count, HAfin_count, *HBfin_count*, *HAmss_data*, *HBmss_data*, HAmaxseg_size, HBmaxseg_size, HAminseg_size, HBminseg_size, HAavgseg_size, HBavgseg_size, HAwin_max, HBwin_max, HAwin_min, HBwin_min, HBwin_zero_ct, HBinitialwin_bytes, HBinitialwin_segs, HAidletime_max, HBidletime_max, HAthroughput, HBport

우를 따로 분리한 ROC 커브이며, 분류율과 오탐율이 윈도우 크기가 4일 때가 1일 경우보다 좋음을 알 수 있다. 이는 제안한 방법이 기존 방법에 비해 탐지 및 분류율을 개선시켰음을 보여준다.

[표 6]은 거리 1일 때, 윈도우 크기 1과 4의 선택된 척도를 비교이다. 윈도우 크기 1에 비해 윈도우 크기 4



〔그림 6〕 거리 1일 때, 분류율과 오탐율

는 수신 ACK 패킷 개수를 나타내는 척도인 *HAdsack_sent*와 데이터 처리량을 나타내는 척도인 *HBthroughput*가 선택되지 않았지만, [그림 6]을 통해 결과가 좋음을 알 수 있다. 윈도우 거리가 증가함으로써 척도가 감소했지만 높은 분류율과 오탐율을 보이는 것은 80개의 모든 척도가 탐지에 도움이 되지 않는 것을 보여주며, 척도의 선택에 따라 결과에 영향을 미치는 것을 알 수 있었다. 그리고 척도가 감소함으로써 연산속도와 정확한 규칙 생성에 도움을 준다는 것을 알 수 있었다.

V. 결 론

본 논문에서는 네트워크 상의 악성 코드 데이터를 추출하고 분석함으로써 특징을 찾아내어 탐지에 유용한 척도를 선정하고 탐지규칙을 생성했다. 단순한 패킷 정보가 아닌 TCP 연결을 구성하는 80가지의 통계적 정보들을 척도로 추출하여 히스토그램 방법을 이용하여 각 공격의 각 척도 별로 확률 분포 값을 구한다. 그리고 단순 추정량을 적용하여 히스토그램 방법에서 임의로 나눠진 구간에 의해 오분류 될 수 있는 단점을 개선하였으며, 상대 복잡도를 이용해서 거리에 따른 유용한 척도를 선택했다. 이 방법은 페이로드를 분석 대상에서 제외하고 TCP/IP의 헤더 정보만을 이용하여 탐지함으로써 탐지부하를 최소화한다.

선정된 척도를 바탕으로 의사결정트리 알고리즘 중 하나인 C4.5에 적용하여 탐지규칙을 생성하고 자동으로 탐지규칙으로 변환하여, 악성코드에 대한 탐지를 실험했다. 본 실험에서 분류 및 탐지 결과는 거리와 윈도우 크기에 따라 달랐으며, ROC 커브로 실험결과를 보임으로써 제안된 방법이 유용함을 보였다. 그리고 유용

한 탐지척도의 선정이 생성된 탐지규칙의 정확성과 탐지 효율성 및 연산 시간에 영향을 미치는 것을 확인할 수 있었다. 향후 연구에서는 더 다양한 악성코드 데이터를 확보하고 척도를 선정하여 실험하고자 한다.

참고문헌

[1] Hyang-Ah Kim and Brad Karp, “Autograph: Toward Automated, Distributed Worm Signature Detection,” in the Proceedings of the 13th Usenix Security Symposium (Security 2004), 2004.

[2] Xuan Chen and John Heidemann, “Detecting Early Worm Propagation through Packet Matching,” Technical Report ISI-TR-2004-585, 2004.

[3] M. Bykova, S. Ostermann, “Statistical Analysis of Malformed Packets and Their Origins in the Modern Internet,” 2nd IMW, 2002.

[4] M. Mahoney, P. Chan, "PHAD: Packet Header Anomaly Detection for Indentifying Hostile Network Traffic," Technical Report CS-2001-4, Florida Tech., 2001.

[5] M. Mahoney, P. Chan, "Learning Models of Network Traffic for Detecting Novel Attacks," Florida Institute of Tech. Technical Report CS-2002-08, 2002.

[6] S. Templeton, K. Levitt, “Detecting Spoofed Packets,” Proc. of the DARPA Information Survivability Conferences and Exposition, 2003.

[7] Gil-Jong Mun, Yong-Min Kim, DongKook Kim, BongNam Noh “Improvement of Detection Ability According to Optimum Selection of Measures Based on Statistical Approach,” CISC 2005, pp.254-264, 2005.

[8] Gil-Jong Mun, Yong-Min Kim, DongKook Kim, BongNam Noh, "Network Intrusion Detection Using Statistical Probability Distribution," ICCSA 2006, pp.340-348., 2006.

[9] 정일안, “분류 기법을 이용한 네트워크 공격 탐지 규칙의 자동생성,” 전남대학교, 2004

[10] B. W. Silverman, "Density Estimation for Statistics and Data Analysis," Chapman and Hall, 1986.

[11] W. Lee and D. Xiang, “Information-Theoretic Measures for Anomaly Detection,” in Prorc. of the 2001 IEEE Symosium on Security and Privacy, pp. 130-143, 2001.

[12] Ross Quinlan, J., "Constructing Decision Trees, C4.5: Programs for Machine Learning, 2nd edn.," Morgan Kaufmann, pp. 17-26, 1993.

[13] Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification. 2nd edn.", Wiley-Interscience, pp. 49-50, 2000.

〈著者紹介〉

문 길 종(Gil-Jong Mun) 학생회원

2004년 2월 : 전남대학교 컴퓨터정보학부 학사 졸업
2006년 2월 : 전남대학교 정보보호협동과정 석사 졸업
2006년 3월~현재 : 전남대학교 정보보호협동과정 박사과정
<관심분야> 네트워크 보안, 침입탐지, 데이터 마이닝, 정보보호 등

김 용 민 (Yong-Min Kim) 종신회원

2002년 2월 : 전남대학교 전산통계학과 박사 졸업
2004년 3월 ~ 2006년 2월 : 여수대학교 정보기술학부 전임강사
2006년 3월 ~ 현재 : 전남대학교 문화콘텐츠학부 조교수
<관심분야> 시스템 및 네트워크 보안, 전자상거래 보안, 정보보호 등