SVM을 사용한 스캐닝 웜 탐지에 관한 연구

김대공°, 문종섭
고려대학교 정보보호 대학원
arbell@cist.korea.ac.kr°, jsmoon@korea.ac.kr

An Approach for Scanning Worm Detection using SVM

Daegong Kim°, Jongsub Moon
Center for Information Security Technology, Korea University

요 약

기존의 웜 탐지는 중요한 네트워크 소스의 폭주와 스위치, 라우터 및 말단 시스템에서의 변동 효과를 가지고 공격을 판단했었다. 하지만 최근의 인터넷 웜은 발생 초기에 대응하지 못하면 그 피해의 규모가 기하급수적으로 늘어난다. 또한 방어하기가 어려운 서비스 거부 공격을 일으킬 수 있는 간접 공격의 주범이 될 수 있다는 점에서 웜의 탐지와 방어는 인터넷 보안에 있어서 매우 중요한 사안이 되었다.

본 논문에서는 이미 알려진 공격뿐 아니라 새로운 웜의 스캐닝 공격을 탐지하기 위하여, 패턴 분류 문제에 있어서 우수한 성능을 보이는 Support Vector Machine(SVM)[1]을 사용하여 인터넷 웜의 스캐닝 공격을 탐지하는 시스템 모델을 제안한다.

1. 서 론

컴퓨터 바이러스 및 인터넷 웜의 기하급수적인 증가와 공격 기술의 발달로 공격에 대한 탐지 및 방어는 날이 갈수록 어려워지고 있다.[2]

웜의 기원을 살펴보면 1988년에 나타난 Morris 웜이 그 시초이다. 그 후, 수많은 웜이 나타났고 그중 Code-Red와 Nimda는 전 세계적으로 10만대 이상의 컴퓨터를 감염시켰었다. 그리고 2003년에 들어서 SQL-Slammer[3]웜의 등장은 취약한 호스트(75,000)의 90% 이상을 10분 안에 감염시키는 능력을 발휘하였다.

최근의 웜인 Witty[4]의 경우 SQL Slammer/Sapphire 웜과 같은 다른 웜 보다 상대적으로 매우 빠르게 전파되어 사람이 대응하기도 전에 이미 감염된다. 이러한 공격의 변화는 사람이 공격의 타입에 따른 초기 대응을 직접할 수 있는 시간적인 여유가 없다는 것을 명백하게 알려주었다. 따라서 스캐닝 웜 공격의 자동화된 탐지와 방어 메커니즘이 절실하게 요구 되었다.

웜은 타겟을 어떻게 찾는지에 따라 다음의 타입으로 분류된다.

가) 위상적인 웜 : 새로운 타겟에 대한 정보를 감염된 시스템에 저장된 데이터에서 찾는다.

나) 수동적인 웜 : 타겟을 적극적으로 찾지 않으며, 다른 곳으로 전파되기 위해 사용자의 행위에 의지한다.

다) 스캐닝 웜 : 새로운 타겟을 찾기 위해 IP 주소 블록을 사용하여 순차적으로 스캐닝 하거나 무작위 방법을 사용하여 주소를 선택한다.

본 연구는 Code-Red, SQL-Slammer 및 Witty와 같은 스캐닝 웜에 초점을 두고 있다.

2. 관련 연구

인터넷 웜 공격을 탐지, 대응, 방어하기 위해서 다 수의 방법들이 제안되었다. 웜을 봉쇄하는 시스템은 주소 블랙리스팅(address blacklisting)과 내용 필터링(content filtering) 두 가지 분류로 나눌 수 있다. 주소 블랙리스팅은 확실한 네트워크 주소에서의 부정행위를 탐지하고 그들로부터의 어떤 접근 시도도 막는 방법이다. 내용 필터링 방법은 웜의 네트워크 연결 특징을 확인하고 나서 그 특징들과 같은 모든 커넥션을 필터링한다. Virus Throttle 방법이 첫 번째 타입의 예이고, Packet Matching과 DAW 방법이 두 번째 타입이다.

Virus Throttle 방법[5]은 취약한 호스트를 감염시키기 위해 웜 스캐닝 시 동시에 많은 호스트로의 통신을 수반한다는 사실을 근거로 하여 Williamson에 의해 제안되었다. Packet Matching 알고리즘[6]은 Xuan Chen과 John Heidemann에 의해 제안되었다. 이 알고리즘은 웜이 보통 몇몇 특정 보안 취약점에 상응하는 지정된 포트 번호를 이용한다는 사실에 근거를 두고 있다. 그리고 웜의 랜덤 스캐닝이 정상적인 호스트들의 행동 보다 높은 실패율로 나타나는 것에 근거를 두는 Distributed Anti-Worm 기술이(DAW)[7] 제안되었다.

이외에 다른 접근으로는 Worm의 다형성을 처리하기 위해서 Worm의 모델을 미리 분석하고 웜 공격을 탐지하는 Colored Petri-net(CPN) 기술을 이용한 Worm 행위를 감시하는 연구[8]가 있다. 그리고 웜의 스캐닝이 라우터에서 네트워크 프로토콜 운영상에 2차적인 자원 소모효과를 가져온다는 사실에 근거를 두어 시뮬레이션을 통한 웜 탐지에 관한 연구[9]도 이루어지고 있다.

3. 웜 탐지 방안

3.1 웜 탐지를 위한 SVM

인터넷 웜을 탐지하는 여러 가지 방법들 중 시그니처 방식이 아닌 새로운 웜에 대한 탐지가 가능한 anomaly 탐지 방법들을 앞에서 언급하였다. 탐지 방법들을 살펴보면 웜을 탐지하기 위하여 탐지 방법들 마다 threshold 값을 정한 것을 알 수 있다. Virus Throttle 방법에서는 시간과 연결 시도 횟수가 threshold 값이 되며, CounterMalice 방법에서는 점수의 기준값, Packet Matching 방법에서는 민감도, DAW 방법에서는 실패율 λ 가 threshold 값이 된다. 이러한 threshold 값은 환경에 따라 큰 차이를 보일 수 있으며 그때 마다 threshold 값을 지정해 주어야 한다. 또한, 대부분 if-then 방식의 접근을 보이고 있어 예측할 수 없는 다양한 환경에서 웜 탐지를 우회하는 공격에 쉽게 노출 될 수 있다.

본 논문에서는 이러한 threshold 값을 지정하는 수고 없이 Support Vector Machine(SVM)을 이용하여 웜 공격과 정상 트래픽을 구분 탐지하는 방법을 제안한다.

SVM 방법은 원래 두 개의 그룹이 있는 경우에 두 그룹간의 경계선(decision boundary)을 매우 효과적으로 도출해 내는 분류방법으로서, 판별오류를 최소화하면서 두 그룹을 가장 선명히 분리시키는 hyperplane을 찾고자 하는 시도이다 (Vapnik 1996; Burges 1998). 그룹경계선은 주로 hyperplane으로 나타내며 선형일수도 있고 비선형일수도 있다.

SVM을 이용하여 네트워크의 트래픽을 웜 공격 트래픽과 그렇지 않은 두 분류로 나누는 hyperplane을 발견하는데, 웜 공격이 이루어지는 traning data set을 적용하여 hyperplane을 결정하고 test data set으로 탐지 효율을 측정한다.

3.2 웜 탐지 모델

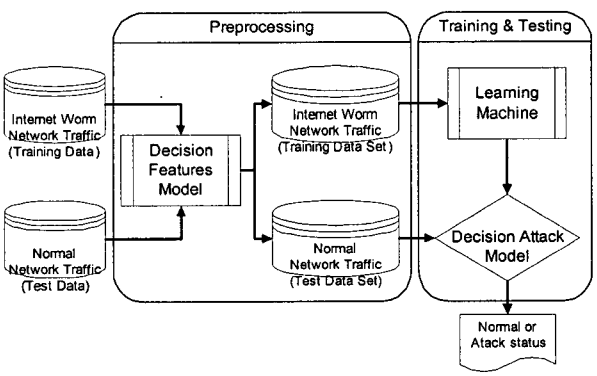

그림 1 SVM 학습을 통한 웜 탐지 모델

SVM 학습을 통한 스캐닝 웜 탐지방안의 구조는 [그림 1]과 같다. 먼저 학습과 테스트를 위한 패킷을 수집하고, 수집된 트래픽 데이터를 전처리 과정을 거쳐 feature들을 결정하여 Training Data Set과 Test Data Set을 구성한다. Training Data Set은 SVM의 학습 데이터로 사용되고 Test Data Set은 공격 행위에 대한 탐지 효율을 측정하는데 사용된다.

4. 실 험

4.1 Training Data & Test Data

실험에 사용할 Training Data와 Test Data는 SVM에 적용하여 훈련 및 테스트하기 전에 Decision Features Model에서 전처리 작업된다.

실험을 위하여 먼저 SVM을 학습 및 테스트시키기 위한 Training Data와 Test Data를 구성하였다. 이상탐지의 특성상 학습 시 사용되지 않은 데이터로 테스트를 하는 것이 좀더 정확한 결과를 얻을 수 있으므로, 정상 데이터의 경우 학습용 패킷과 테스트용 패킷을 다르게 하여 데이터를 수집하였다.

Training Data는 모두 10000개의 패킷으로 만들어 졌으며 각각 정상 패킷 5000개와 웜 스캐닝 패킷 5000개로 이루어져 있다. Test Data는 모두 4000개의 패킷으로 학습 시 사용되지 않았던 정상 패킷 1000개와 웜별 스캐닝 공격 패킷을 각각 1000개씩 가지고 있다. Traffic Data 구성을 표로 나타내면 다음과 같다. (단, Transport protocol only)

표 1 Training & Test Data

구분	Training Data	Test Data
정상 데이터	TCP/IP packet (5000)	TCP/IP packet (1000)
공격 데이터	Worm.Code-Red	Worm.Code-Red packet (1000)
		Worm.SQL.Slammer packet (1000)
		Worm.W32.Witty packet (1000)
Total	10000	4000

(단위: 개)

4.2 Decision Features

SVM에 적용되어질 Data Set을 만들기 위하여 웜의 Feature를 선정하는데 Georgia Tech Net work Simulator(GTNetS)[10]에서 연구된 인터넷 웜 모델을 사용하였다. 이 모델은 인터넷 웜의 시뮬레이팅에 관한 연구이며, 웜의 행위를 지정하는 다수의 파라메터를 정의하고 있다. 본 논문의 실험에서는 인터넷 웜 모델에서 제시한 여러 파라메터 중 다음과 같은 파라메터를 Feature로 사용한다.

표 2 Features Table

Mode	Feature Category	Feature 정의
A	Transport protocol	Protocol field
	Infection length	Len field
	Target vector	S-IP address field D-IP address field
B (A 포함)	Scan rate (UDP)	S-IP, D-IP 개수(10s)
	Connections (TCP)	S-IP, SYN 개수(10s)

Features Table의 Mode-B는 Transport 프로토콜의 타입에 따라 Feature를 다르게 구한다.

4.3 실험 결과

SVM을 이용한 웜 탐지 실험은 커널 함수로서 linear와 polynomial을 사용하여 각각 진행하였다.

표 3 SVM linear 커널 함수를 이용한 테스트 결과

Kernel function	linear			
Detection rate	false positive		false Negative	
Mode	A	B	A	B
Code-Red	6.3	1.8	46.2	24.8
SQL-Slammer	7.9	3	88	53.2
Witty	7.1	2.5	82.1	49.6

표 4 SVM polynomial 커널 함수를 이용한 테스트 결과

Kernel function	polynomial			
Detection rate	false positive		false Negative	
Mode	A	B	A	B
Code-Red	3.8	0.1	0	0
SQL-Slammer	4.7	1.7	49.3	32
Witty	4.2	0.6	41.9	24.8

실험 테스트 결과 전체적으로 linear 커널 함수보다 polynomial 커널 함수에서 상대적으로 나은 탐지 효율을 보였다. 또한, Mode-A Features Set에서의 결과보다 Scan rate와 Connections 값이 추가된 Mode-B Features Set에서 확연히 나은 성능을 보였다. 실험을 통하여 탐지 효율은 커널의 커스터마이징과 Features Set의 구성이 가장 큰 요인으로 작용된다는 것을 알 수 있다. 특히, Mode-B에만 포함된 타임스탬프에 의한 스캐닝 속도는 웜 탐지에 큰 영향을 미치는 Feature임을 알았다.

5. 결 론

본 논문에서는 SVM을 이용한 스캐닝 웜 탐지 방안을 제안하였다. 비록 웜의 스캐닝 공격으로 한정하였지만 최근 주로 사용되는 웜 공격 타입인 스캐닝 웜을 사용하여 실험하였다. 실험에서와 같이 미리 학습된 공격 기법은 테스트 시 거의 정확하게 탐지하며 학습을 하지 않았던 웜도 일정부분 탐지하는 것을 알 수 있었다. 이는 기존의 웜 이외에 새로운 형태의 웜 공격에 대해서도 탐지할 수 있음을 나타낸다. 또한, 커널의 커스터마이징과 최적의 Features 선정으로 탐지 효율을 높일 수 있다는 것을 알 수 있었다.

향후 랜덤 트리 토폴로지의 링 모델[11]등을 사용하여 다양한 네트워크상에서의 웜 공격 탐지 효율에 관한 연구가 필요하다. 더 나아가 성능 향상 변수로 사용될 수 있는 여러 가지 커널의 도입과 탐지 효율을 높일 수 있는 다양한 Feature들을 연구함으로써 웜 탐지분야에 한 축을 세울 수 있을 것으로 사료된다.

참고 문헌

[1] Ruping S, mySVM-a Support Vector Machine, University of Dortmund, 2004.

[2] Q.S.Zhao, Research on and Enforcement of Malware-Defending Technology of Secure Operation System, Ph.D. thesis, Beijing, Institute of Software Chinese Academy of Sciences, 2002 (in Chinese).

[3] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, Inside the slammer worm, IEEE Magazine of Security and privacy, 1(4),33,39, July/August 2003.

[4] Symantec Security Response, W32.witty.worm, http://securityresponse.symantec.com/avcenter/venc/data/w32.witty.worm.html, January 2004.

[5] J. Twycross, M. M. Williamson, Implementing and testing a virus throttle, In Proceedings of the 12th USENIX Security Symposium, 285, 294, Washington, D.C., USA, August 2003.

[6] X. Chen, J. Heidemann, Detecting early worm propagation through packet matching, Technical Report ISITR-2004, 585, USC/Information Sciences Institute, February 2004.

[7] S. Chen, Y. Tang. Slowing down internet worms, In Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04), Tokyo, Japan, March 2004.

[8] Liu Peishun, Wang Jianbo, He Dake, Worm detection using CPN, IEEE International Conference on Systems, Man and Cybernetics, 2004.

[9] Ihab Hamadeh, Jason Hart, George Kesidis, Venkat Pothamsetty, A Preliminary Simulation of the Effect of Scanning Worm Activity on Multicast, Proceedings of the Workshop on Principles of Advanced and Distributed Simulation (PADS'05), 2005.

[10] G. F. Riley, The Georgia Tech Network Simulator, In Proceedings of the ACMSIGCOMM workshop on Models, methods and tools for reproducible network research, 5, 12, ACM Press, 2003.

[11] M. I. S. George F. Riley, W. Lee, Simulating internet worms, In Proceedings of The IEEE Computer Societys 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'04), 268, 274, Volendam, The Netherlands, October 2004.