IDS 성능 향상을 위한 DEVS 모델링*

DEVS Modeling for IDS Performance Improvement

서희석** , 조대호**
Seo, Heesuk , Cho, Taeho
**성균관대학교 전기 전자 및 컴퓨터 공학부

Abstract

침입 탐지 시스템이 침입 행위를 탐지하기 위해서 시간에 대한 처리를 고려하지 않고는 침입을 탐지할 수 없는 경우(예 Denial of Service)가 존재한다. 즉 사건의 발생 시점에 대한 처리 없이는 침입 탐지가 불가능하다. 본 논문에서는 시뮬레이션 모델을 통하여 시간에 관한 처리를 체계적으로 구성하고, 여러 가지 상황을 조성하여 반복적으로 실행함으로써 침입 탐지 시스템의 핵심 요소인 침입 판별을 효과적으로 수행할 수 있도록 하였다.

1 서론

e-business의 영역이 확대되면서 사이버 공간에 대한 각종 침해 사고가 빈번하게 발생하고 있다. 이런 침입을 효율적으로 대처하기 위해 보안 분야에 대한 중요성이 대두되고 있다. 지난 2월 Yahoo, EBay, Amazone, CNN등의 web이 해킹 당함으로 기업 이미지는 물론 재정적으로 엄청난 손실을 입은 것을 본다면 인터넷의 발전과 보안의 발전이 얼마나 밀접한 연관을 맺는가를 확인할 수 있다.

네트워크의 속도가 급속하게 발전하는 상황에서 많은 양의 데이터를 처리해야하는 보안 시스템을 직접 사용해 성능을 평가하는 것은 효율적이지 못하다. 본 연구를 위해서 DEVS(Discrete Event System Specification) 방법론을 이용하여 침입 탐지 시스템을 테스트할 시뮬레이션 환경을 구축하였다. 시뮬레이션 모델은 추상화 과정을 거쳐 완성되는데, 이런 모델에 사용되는 입력 및 출력도 추상화하여 사용하는 것이 일반적이다. 하지만 본 시스템에서는 네트워크에서 수집한 실 패킷(real packet)을 시뮬레이션 모델의 입력으로 사용하여 실제 시스템에 가깝도록 구성하였다.

*본 연구는 bk21 사업에서 진행되는 연구임.

2. 배경 이론

2.1 DEVS 방법론

Zeigler에 의해 정립된 DEVS 방법론은 연속적인 시간상에서 발생되는 이산 사건을 처리하는 시스템을 시뮬레이션하기 위해 이론적으로 정립된 모델링 방법론이다[1][2]. 이는 모델의 구조와 행동을 시뮬레이션 수행으로부터 추상화시키기 위해 모델을 집합 이론적 방법으로 이용한 것으로, 시스템을 계층적(hierarchical)이고 모듈화(modular)된 형식으로 기술한다.

DEVS에서는 기본(Basic) 모델과 결합(Coupled) 모델을 정의한다. 기본 모델은 시스템의 동적인 특성을 표현하기 위한 것이고, 결합 모델은 시스템의 구성 요소간에 상호 작용을 표현하기 위한 것으로 다음과 같은 항들로 명세 할 수 있다.

$M = \langle X, S, Y, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, t_a \rangle$
- X : 입력 사건의 집합
- S : 상태들의 집합
- δ_{int} : 내부 상태 변이 함수
- δ_{ext} : 외부 상태 변이 함수
- λ : 출력 함수
- t_a : 시간 갱신 함수

$DN = \langle D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, select \rangle$
- D : 구성 요소 이름의 집합
- M_i : 구성 모델
- I_i : 모델 i와 연관된 모델의 집합
- $Z_{i,j}$: 모델 i와 j모델간의 연결 함수
- *select* : *tie-breaking selection* 함수

2.2 침입 탐지 시스템

침입 탐지 시스템(IDS : Intrusion Detection System)은 외부의 침입에 대해 능동적으로 대처하는 시스템으로 방화벽(firewall)과 함께 활용되는 네트워크 보안 솔루션이다[3][4][5]. 방화벽은 단순히 네트워크를 통한 외부 침입을 차단하지만 침입 탐지 시스템은 방화벽의 앞 또는 뒤에서 침입 사실을 탐지해 침입자의 공격에 대응하기 위한 솔루션이다.

침입 탐지 시스템은 크게 데이터 소스(data source)를 기반으로 분류하는 방법과 침입 탐지 모델을 기반으로 분류하는 방법으로 나눌 수 있는데 데이터 소스를 기반으로 분류하는 방법은 파일 점검 및 각 어플리케이션들의 버전 확인 등을 통한 취약점 점검들을 실행함으로 시스템의 불법 사용에 대한 점검과 예방을 목적으로 한다. 이에는 단일 호스트 기반의 침입 탐지 시스템, 다중 호스트 기반의 침입 탐지 시스템, 네트워크 기반의 침입 탐지 시스템이 있다. 침입 모델을 기반으로 분류하는 방법은 컴퓨터 자원의 비정상적인(anomalous) 동작이나 사용에 근거한 침입 탐지와 정해진 모델을 벗어나는 경우를 침입으로 간주하는 비정상적인 침입 탐지 기법(Anomaly Detection Technique)과 Auditing 정보를 활용하여 시스템이나 응용 소프트웨어의 약점을 통하여 시스템에 침입할 수 있도록 잘 정의된 공격 형태나 침입이라 규정해 놓은 정해진 모델과 일치하는 경우를 탐지하는 오용 침입 탐지 기법(Misuse Detection Technique)으로 나눌 수 있다.

2.2.1 침입 탐지 시스템의 신뢰성 고찰

침입 탐지 시스템의 핵심 부분은 침입을 분석하고 탐지하는 부분이다. 본 연구에서는 이 부분을 AGENT 모델이 담당하도록 되어있는데, 각 AGENT 모델은 내부에 전문가 시스템을 내장하고 있어 전문가 시스템의 규칙(Rule)을 이용하여 침입을 탐지하게 구성되었다.

침입 탐지 시스템에 분류할 수 있는 2가지 오류가 있는데 바로 false positive와 false negative이다. false positive는 침입이 아니지만 침입 탐지 시스템이 침입으로 오진하는 오류이고, false negative는 침입이 분명한데 침입 탐지 시스템이 침입으로 간주하지 않는 오류이다. 시스템 관리자는 침입 탐지 시스템을 운용함에 있어서 이 두 가지 오류를 고려하지 않으면 안 된다. 시스템을 외부로부터 안전하게 보호하기 위해서 침입 탐지 시스템에서 사용되는-정확하게 전문가 시스템의 규칙에서 사용되는- 임계값(threshold)을 엄격한 수준으로 설정하게 된다면 false positive 오류가 증가하게 되고, 너무 느슨하게 임계값을 관리하게 된다면 분명한 침입을 판별하지 못하는 false negative의 증가를 가져오게 된다. 이 두 가지는 항상 trade off 관계를 갖기 때문에 관리자의 정책에 의한 적정한 임계값 설정이 중요한 사항으로 대두된다.

1) 임계값에 영향을 줄 수 있는 요인
- 관리자의 시스템에 대한 정책
- 시스템의 보안 수준
- 운영체제의 특성
- 시스템의 성능(CPU, memory ....)
- 네트워크의 속도 및 구성
- 기타

2) 규칙과 임계값의 고찰

전문가 시스템의 규칙(Rule)은 전문가의 지식을 담고 있는 곳이다. 이 규칙에서 침입을 탐지하기 위해 임계값을 참조하여 각각의 공격을 판별하게 구성되었다. 1)의 사항들에 의해 임계값이 결정되는데, 만약 1)의 사항을 규칙으로 표현해서 적용할 수 있다면 즉 임계값의 일부를 규칙으로 구성할 수 있다면 좀 더 유연하게 임계값을 정할 수 있을 것이다.

3. 침입 탐지 모델

3.1 침입의 분류

침입 탐지 모델을 구성하기 위해서 G 모델(Generator model)에서 발생된 패킷을 사용하는 용도에 따라 [표 1]과 같이 3가지로 분류하였다. 하나의 패킷을 분석해 침입을 판정할 수 있는 경우는 네트워크에서 수집된 패킷의 헤더 정보 중 하나 혹은 그 이상의 플래그가 정상적이지 못한 경우이다. 다수 개의 패킷을 분석해서 침입을 판정할 수 있는 경우는 여러 개의 패킷을 검사해야 침입 탐지할 수 있는 경우로써 시스템을 시뮬레이션 모델로 구성할 경우의 이점이 여실히 증명될 수 있는 부분이다. 수집된 패킷의 데이터 영역을 확인함으로 침입을 판정할 수 있는 경우는 패킷의 헤더의 정보를 통해서 공격을 확인할 수 없는 경우이다. 즉 정상적인 네트워크의 연결을 가장해 시스템 관리자의 권한을 획득하기 위해 일련의 활동을 하는 경우에 해당된다.

3.2 EF-IDM 모델 디자인

침입 탐지 시스템을 시뮬레이션 모델로 구성했을 경우의 장점은 각 단계(모델)에서 특별한 시간에 대한 처리 없이, 모델이 시간을 사용할 수 있다는 것이다. DoS의 경우는 짧은 시간 동안에 많은 양의 네트워크 트래픽을 발생시켜 시스템을 다운시키거나 다른 클라이언트 시스템이 서버에 접속하지 못하도록 공격하는 기법인데, 이러한 공격은 시간에 대한 처리를 하지 않고는 해결할 수 없다. 그러므로 시뮬레이션 모델을 사용해 각 모델에서 시간을 다룰 수 있도록 구성한다면 효과적으로 침입을 탐지할 수 있다[3][5][6].

EF-IDM 모델은 크게 EF 모델, IDM 모델로 구성되었고, IDM 모델은 내부에 PCL 모델과 AGENT 모델을 포함한다. [그림 1]은 EF-IDM의 모델 구성도이다.

3.2.1 EF 모델

EF(Experimental Frames) 모델은 네트워크에서 수집된 패킷을 수집된 시간 간격에 맞도록 생성하기 위해 구성된 G 모델(Generator model)과 통계적 처리를 위해 생성된 패킷과 처리된 패킷을 저장하고 있는 T 모델(Transducer model)로 구성된다.

G 모델은 네트워크에서 수집된 패킷을 시뮬레이션 수행을 위해 생성한다. 시뮬레이션은 추상

[표 1] 패킷의 분류

	하나의 패킷을 분석	다수의 패킷을 분석	패킷의 데이터를 사용
공격 형태	-Probing · Port Probing · Protocol Probing -Denial of Service · WinNuk · X-mas · ping of Death · Land Attack	-Scan · Port Scan · Address Scan · CGI-Query -Denial of Service · ICMP Flood · Smurf · Web-Port DoS · Mail-Port DoS · DNS DoS · Mail-Bomb · UDP Bomb · SYN Flood	· Door Knob Rattling · Password Cracking · Buffer overrun · Environment Variable overflow Attack · 침입에 자주 사용되는 명령어 사용 · 관리자만 사용할 수 있는 명령어 사용

화 과정을 거쳐 모델을 구성하고, 모델에서 사용될 입력도 추상화를 거쳐 사용하는 것이 일반적이지만 EF-IDM 모델에서는 최대한 실제 침입 환경과 비슷하게 구성하기 위해서 실제 침입을 발생시킨다. G 모델은 이런 실제 패킷을 생성하는 것이다.

3.2.2 IDM 모델

IDM(Intrusion Detection Model)은 실제 침입을 탐지하는 모델로 PCL 모델과 AGENT 모델을 포함한다.

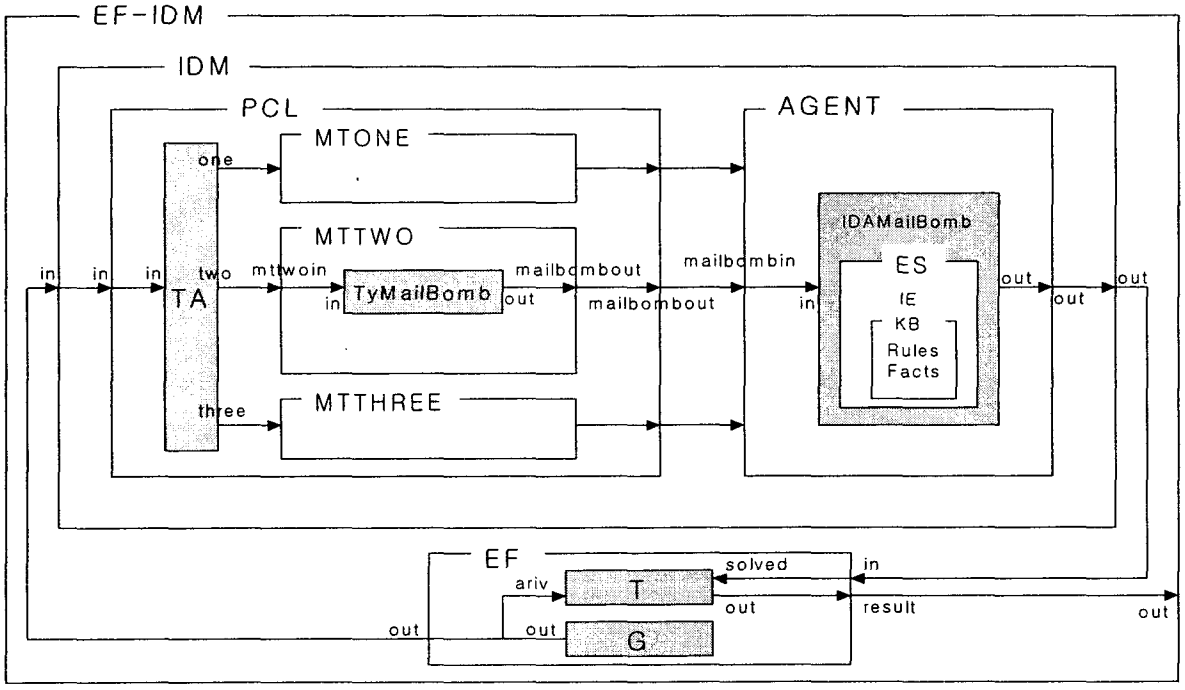
1) PCL(Packet Classify Library) 모델은 G 모델에서 생성된 패킷을 입력으로 받아서 2.2에서 분류한 각 종류에 맞게 패킷을 분배하고 에이전트가 패킷을 사용할 수 있도록 축약하는 역할을 담당한다.

우선 G 모델에서 생성된 패킷은 TA(Task Allocator) 모델이 받아서, 3종류의 모델 즉 MTONE, MTTWO, MTTHREE 모델에게 분배하게 된다. 그러면 각 모델은 각 공격에 해당하는 모델(예 : TyMailBomb, TyPortScan....)에게 패킷을 전달하고, 각 공격에 해당하는 모델은 패킷의 정보를 확인해서 자신에 해당하는 공격이라면 에이전트에게 넘기고, 의미 없는 패킷이라면 패킷을 소멸시킨다.

EF-IDM 모델의 테스트를 위해서 TyMailBomb 모델과 TyPortScan 모델이 MTTWO 모델에 기본 모델로 탑재되어 있다.

침입 탐지를 위해 MTONE, MTTWO, MTTHREE 모델을 확장하기 위해서는 침입에 해당되는 기본 모델을 구성하여 결합 모델인 MTONE, MTTWO, MTTHREE 모델에 첨가하면 된다.

2) AGENT 모델은 침입 탐지의 핵심 모델로 전문가 시스템(ES : Expert System)을 내장하고 있다. Mail-Bomb 공격의 경우 PCL 모델의 TyMailBomb 모델에서 AGENT 모델의 IDAMailBomb 모델로 패킷이 전달되는데,



[그림 1] EF-IDM의 모델 구성도

IDAMailBomb 모델은 이 패킷을 입력으로 받아서 전문가 시스템이 사용할 수 있는 사실의 형태로 전환해 전문가 시스템에게 사실을 전달하게 된다. 전문가 시스템은 IDAMailBomb 모델이 넘겨준 사실을 이용해 자신이 갖고 있는 규칙에 적용해 추론을 하게 된다.

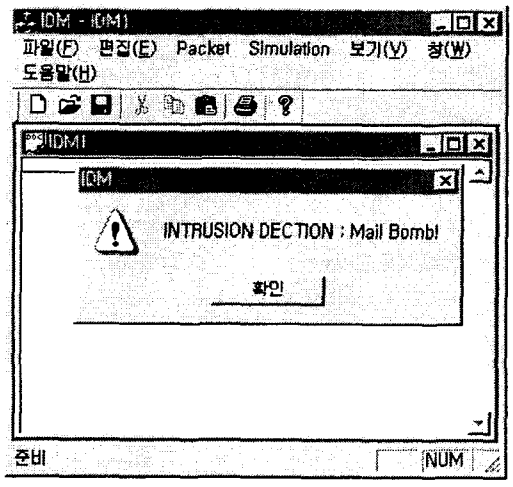
4. 시뮬레이션

시뮬레이션 모델을 사용해 구성된 모델이 실제 침입을 정확히 탐지하는지 시뮬레이션을 수행해 보았다.

4.1 실행

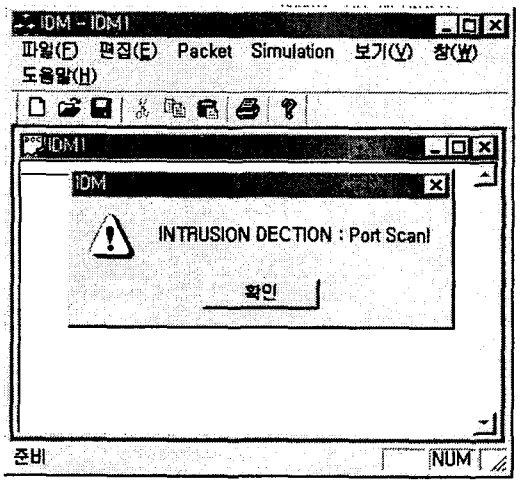
각 공격을 실행하기 위한 시뮬레이션 환경으로는 DEVS-ObjC을 사용하였고, 침입을 위해 Mail-Bomb과 Port Scan 공격을 시도하였다.

Mail-Bomb 공격은 메일 서버에 폭탄 메일을 보내는 DoS 공격의 한 종류로 Mail-Bomb 공격 패킷을 생성하기 위해 KaBoom version 3.0을 사용하였다. [그림 2]는 침입이 탐지되었을 때 화면에 출력되는 창을 보여준다.



[그림 2] Mail-Bomb 공격을 탐지한 경우

Port Scan은 침입을 하기 위해 시스템의 열려진 포트를 찾아보는 공격의 사전 답사로서 Port Scan 공격 패킷을 생성하기 위해서 Peter Harrison이 고안한 Port Scan Plus를 사용하였다. [그림 3]은 침입을 탐지했을 화면에 출력되는 창을 보여 준다.



[그림 3] Port Scan 공격을 탐지한 경우

4.2 결과

침입 탐지 모델이 공격자의 침입을 탐지할 수 있는지 알아보기 위해 Mail-Bomb 공격과 Port Scan 공격을 시도해 보았다. 이러한 시뮬레이션 환경은 관리자가 임계값을 수월하게 조정할 수 있도록 할 뿐 아니라, 전문가 시스템의 규칙을 추가하고 삭제할 경우 규칙에 의한 핵심 판결 요소의 설계를 위해 유용하게 사용될 수 있다.

본 연구에서 Mail-Bomb 공격을 탐지하기 위해 규칙에서는 TCP Protocol을 사용한 패킷이 25번 포트로 들어오고, 1초 동안 10개 이상의 패킷이 20초 이상 계속되는 경우를 침입으로 간주하도록 하였다. 여기서 10개, 20초라는 임계값의 값을 변화시킴으로 관리자는 시스템에 알맞은 임계값을 정할 수 있으므로 효과적으로 공격에 대응할 수 있다. 또한 좀더 정교한 탐지를 위해서 규칙을 구성하여 추가하면 더 나은 결과를 얻을 수 있다. 예를 들어 분산 환경에서 Mail-Bomb 공격이 있는 경우에 각각의 공격 시스템은 메일 서버에 연결 설정 패킷을 보내 연결을 설정한 후 폭탄 메일을 보내 공격을 시도하게 된다. 그러므

로 규칙을 구성할 때에 SYN 플래그를 조사하는 규칙을 추가하여 전문가 시스템을 구성하면 되므로 현재와 같은 시뮬레이션 환경은 규칙의 추가, 삭제를 용이하게 할 수 있다.

4. 결론 및 향후 과제

지금까지 침입 탐지 시스템의 성능을 평가하기 위해서 시뮬레이션 모델을 통해 성능을 분석하는 방법을 제시하였다. 실제 시스템을 시뮬레이션 모델로 구성하면서 발생할 수 있는 신뢰성 문제를 최소화하기 위해서 모델에서 사용되는 입력을 실제 패킷을 직접 사용하였다. 또한 실제 활용할 수 있는 침입 판별 요소를 갖는 시뮬레이션 환경을 구축하였다. 침입 탐지를 위해 시뮬레이션 모델을 구성하고, 이런 모델을 통해 여러 가지 상황을 조성하고, 반복적으로 실행함으로써 효과적으로 침입 탐지의 성능을 살펴볼 수 있음을 보였다.

향후 연구 과제로는 좀 더 일반적인 시뮬레이션 환경을 구축하기 위한 침입 생성 모델이 필요하고, 각 시스템에 적합한 임계값을 제시할 수 있는 시뮬레이션 환경을 구축해야 할 것으로 여겨진다.

참고 문헌

[1] Bernard P. Zeigler, "Object-Oriented Simulation with Hierarchical, Modular Models", Academic Press, 1990.

[2]Bernard P. Zeigler, "Theory of Modelling and Simulation", John Wiely, 1976, reissued by Krieger, Malabar , 1985.

[3]Rebecca Gurley Bace, "INTRUSION DETECTION", Macmillan Technical Publishing, 2000.

[4]Denning Dorothy, "An Intrusion Detection Model", Proceedings of the Seventh IEEE Symposium on Security and Privacy. May 1986 : 119-131.

[5]S. Northcutt, "Network Intrusion Detection An Analysts Handbook", New Riders Publishing, 1999.

[6]Edward G. Amoroso, "Intrusion Detection - An Introduction to Internet Surveillance, Correlation, Traps, Trace Back, and Response" Intrusion.Net Books, 1999.

[7]Stuart Mcclure, Joel Scambray, George Kurtz, "Hacking Exposed: Network Security Secrets and Solutions," McGraw-Hill, 1999.

[8] Behrouz A. Forouzan, "TCP/IP Protocol Suite," McGrawHill, 2000.

[9]Ulf Lindqvist, Philip A. Porras, "Detecting computer and Network Misuse Through the Production-Based Expert System Toolset(P-BEST)", In Proceedings of the 1999 IEEE Symposium on Security and Privacy, Oakland, California, May 9-12, 1999.

[10] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network Intrusion Detection", IEEE Network, Vol. 8, No. 3, pp26-41, May/June 1994.

[11]J. Balasubramaniyan, J. Garcia-Fernandez, D. Isacoff, E. Spafford, Diego Zamboni, "An Architecture for Intrusion Detection using Autonomous Agents", Technical Report No. 98-05, COAST Group, Dept. of Computer Science, Purdue University, June 11, 1998.

[12]P. Neumann and D. Parker, "A Summary of computer misuse techniques", In Proceedings of the 12th National Computer Security Conference, October 1989, pp. 396-407.

[13]이미라, "시뮬레이션의 계층적 애니메이션",한국 시뮬레이션 학회 논문지, December 1999.