통합 침입탐지 시스템을 위한 IDEF(Intrusion Detection Exchange Format)의 관리 및 확장에 관한 연구

장지선* 예홍진 조은선

*아주대학교, 정보통신전문대학원

Research on Extension and Management of IDEF for Integrated Intrusion Detection System

Ji-seon Jang* Hong-jin Yeh Eun-sun Cho

*Dep. of Computer and Informaton Engineering in Ajou University

요 약

개개의 침입탐지 시스템에서 탐지한 경보(alert) 자료의 공유를 쉽게 하기 위해서, IETF(Internet Engineering Task Force)에서는 IDEF(Intrusion Detection Exchange Format)모델을 제안하였다[1]. 특히, IDEF는 최근에 관심이 모아지고 있는 다수의 침입탐지 시스템(이하 '통합 침입탐지 시스템')을 통합 관리하는 방식에서 각 단말 침입 탐지 시스템의 침입 경보자료의 수집 관리를 용이하게 할 수 있다. 그러나, 통합 침입탐지 시스템에서 개개의 침입 탐지 시스템에서 발견하지 못하는 침입을 판단하거나 판단의 정확성을 높이기 위해서는 기존의 IDEF에 추가적인 자료가 요구되어 진다. 본 논문에서는 통합 침입탐지 시스템의 상위 시스템에서 수집된 경보 자료를 IDEF의 관계형 데이터베이스 스키마로 변환하는 방식을 제시하였다. 그리고, 통합 침입탐지시스템에서 추가적으로 필요한 자료에 의거하여 DDoS공격탐지에 필요한 자료형을 IDEF에 확장하였다.

I. 서론

침입 탐지 시스템(IDS : Intrusion Detection System)의 주된 임무는 시스템에 침입이 있는가에 대한 판단을 하고 알맞은 침입 경보(alert)를 주는 것이다[2]. 이러한 침입 경보에 이용되는 자료는 여러 침입탐지 시스템마다 다른 형태를 가지고 있어 왔지만, 1998년에 IETF에서 IDEF를 제안함으로써 표준안으로 받아들여지고 있다. 이로써 같은 침입행위에 대해 마치 다른 행위로 인식하는 것을 피할 수 있게 되었으며, 제품들의 성능 비교, 데이터 공유 등을 원활하게 할 수 있게 되었다.

특히, 최근 관심이 모아지고 있는 통합 침입탐지 시스템에서, 각 단말 침입 탐지 시스템의 침입 경보 자료를 상위 시스템이 보고 받을 때 전달되는 자료를 IDEF 형태로 가정한다면, 서로 다른 제작사의 침입탐지 시스템들의 통합을 시도 할 수 있다. 그런데, 통합 침입탐지 시스템에서 개개의 침입탐지 시스템에서 발견하지 못하는 침입을 판단하거나 판단의 정확성을 높이기 위해서는 기존의 IDEF에 정의된 형태의 자료 외에 다른 종류의 자료가 전달되어야 할 필요가 있게 된다.

본 논문에서는 통합 침입탐지 시스템에서 상위 시스템에서 수집된 경보 자료를 관계형 데이터베이스에서 관리 할 수 있도록, IDEF를 관계형 데이터베이스 스키마로 변환하는 방식을 제시하였다. 이로써 관계형 데이터베이스 관리 시스템에서 풍부하게 제공하는 질의(query) 및 기타 기능들을 통하여 수집된 침입 경보에 관한 정보를 효과적으로 분석할 수 있도록 하였다. 그리고, 본 논문에서는 통합 침입탐지시스템에서 추가적으로 필요한 자료에 의거하여 IDEF를 확장하기 위한 시도로서, DDoS공격 탐지에 필요한 자료 형태에 관해 고찰하고 확장된 스키마를 제안하였다. 본 논문에 제안된 방식은 현재 Oracle 8i에서 침입 경보 자료를 관리할 수 있도록 실험되고 있다.

II. 본문

1. IDEF 모델

IDEF를 구성하는 핵심부분은 ALERT 클래스이다. ALERT 클래스는 침입행위가 발생했을 때 경보번호, 발생시간, 시스템에 끼치는 영향 등을 데이터로 저장한다. 그리고, ALERT 클래스와 Inheritance 관계인 클래스로는 TOOLALERT, CORRELATION ALERT, OVERFLOW ALERT가 있다. TOOLALERT 클래스는 침입행위를 발생시킨 공격툴에 대한 정보를 저장하고, CORRELATION ALERT는 침입경보와 관련된 정보를 저장한다. OVERFLOWALERT 클래스는 overflow 공격과 관련된 정보를 저장한다. ALERT 클래스와 Aggregation 관계인 클래스로는 ANALYZER, CLASSIFICATION, TARGET, SOURCE가 있다. ANALYZER 클래스는 침입을 분석하는 개체에 대한 정보를 저장하고 있고, CLASSIFICATION 클래스는 침입정보와 관련된 취약성 정보를 갖고 있다. TARGET 클래스는 침입의 대상에 관한 정보를 갖고 있고, SOURCE 클래스는 침입행위를 일으키거나 일으킬 가능성이 있는 대상에 대한 정보를 저장한다. IETF에서 제안한 데이터 모델을 UML(Unified Modeling Languange)로 표현하면 다음 그림과 같다.

2. 스키마 정의

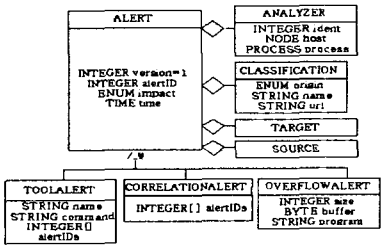

그림 1 : IDEF 데이터 모델

IDEF모델을 Oracle 8i에서 다음과 같이 스키마로 정의하였다.

sn	ver	alertid	impact	time
char(10)	char(5)	char(10)	char(2)	date

표 1 : Type ooalert

asn	ident	host	process
char(10)	char(10)	char(30)	char(30)

표 2 : analyzer class

csn	origin	name	url
char(10)	char(2)	char(30)	char(30)

표 3 : classification class

tsn	targetid	decoy
char(10)	char(2)	char(10)

표 4 : target class

ssn	sourceid	spoofed
char(10)	char(2)	char(10)

표 5 : source class

또, oalert 클래스와 aggregation 관계를 가지고 있는 클래스에 관계를 맺어 주는 테이블을 구현하였다. 각각의 테이블들은 oalert와 analyzer, target, source, classification에 관계를 맺어 주기 위한 필드 값을 foreign 키로 가지고 있다. 그 중 alert 클래스와 analyzer 클래스의 aggregation 관계의 예는 표6과 같다.

sn	asn
foreign key(sn) references oalert(sn)	foreign key(asn) references analyzer(asn)
char(10)	char(10)

표 6 : alert와 analyzer의 aggregation관계

Inheritance 관계를 구성하기 위해서는 이미 만들어 놓은 ooalert라는 object를 이용해야한다. Inheritance는 Oracle 8i 버전부터 제공되어 지는 기능이다[3].

참고문헌에 제시된 방법은 세 가지가 있는데 여기에서는 subtype이 super-type을 포함하는 방법으로 구현해 보았다.

alertt1	name	command	alert2
ooalert type을 상속 받음			
	char(30)	char(30)	char(30)

표 7 : toolalert class

alertt2	alerts
ooalert type을 상속 받음	
	char(30)

표 8 : correlationalert class

alertt3	size	program	buffer
ooalert type을 상속 받음			
	char(10)	char(30)	char(30)

표 9 : overflowalert class

이렇게 구성된 데이터베이스 스키마들에 실제로 질의를 보내어 얻은 결과는 다음과 같다.

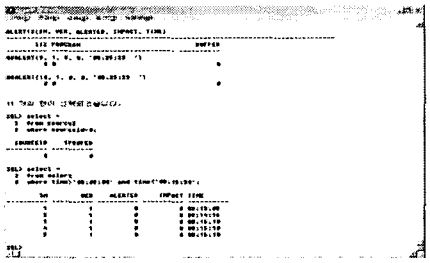

그림 2 : IDEF 스키마의 질의 결과

3. IDEF의 확장 배경

1) 통합 침입탐지 시스템

최근에 개개의 침입탐지시스템에서 발견하지 못하는 침입을 판별해 내기 위해 EMERALD나 GrIDS 같은 통합 침입탐지 시스템에 관심이 모아지고 있다. 그러나, 아직까지 이런 통합 침입탐지 시스템은 IDEF를 활용하지는 못하고 있다[4].

2) DDoS

본 연구에서는 DDoS(Distributed Denial of Service)공격을 탐지하기 위해 IDEF를 이용하고자 한다. 침입 탐지에 대해서 많은 연구가 이루어지고는 있지만, 실제로 DoS공격에 대해서는 아직도 많은 부분이 취약한 현실이다. 이런 DoS공격들은 대다수의 경우 공격을 발생시키는 툴을 이용해서 이루어지고 있다. Stacheldraht를 비롯해서 trinoo나 TFN과 같은 DDoS공격을 유도하는 툴들은 attacker가 master와 agent에 각각 DDoS프로그램을 설치하고, attacker의 명령에 따라 공격을 감행하는 형태를 가진다. 여기에서는 그러한 툴들 중 stacheldracht 툴을 이용한 공격을 중심으로 IDEF를 확장하는 방법을 제시한다.

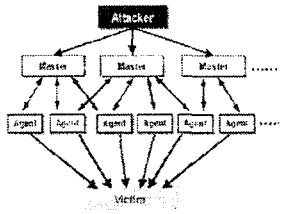

그림 3 : DDoS공격의 구조

3) Stacheldraht의 공격 과정

stacheldraht는 ICMP flood, SYN flood, UDP flood와 Smurf등의 공격에 의해서 DDoS공격을 할 수 있는 기능을 가지고 있다. stacheldraht 네트워크는 1개 또는 그 이상의 핸들러프로그램(mserv.c)과 데몬프로그램(td.c)들의 집합으로 구성된다[5].

각각의 agent에서 데몬이 실행 될 때에는, 어디에 있는 master시스템의 핸들러 프로그램으로부터 지휘를 받아야 할지를 알아야 한다.

agent 데몬이 자신을 지휘할 핸들러들이 결정되면 그 핸들러들에게 ID필드 "666", 데이터 필드에 "skillz"가 포함된 ICMP_REPLY패킷을 보낸다. 핸들러가 그 패킷을 받은 경우는, ID 필드를 667, 데이터 필드에 "ficken"이 포함된 ICMP ECHO_REPLY패킷을 다시 보내준다. 핸들러와 agent 데몬은 주기적으로 666+skillz/667+ficken의 패킷을 보내고 받으면서 서로의 존재를 확인하게 된다.

또, agent 데몬이 실제로 표적 시스템/네트워크에 DDoS공격을 실행할 경우에 표적 시스템 쪽에서 agent의 IP를 추적하지 못하도록 source 주소를 위장한 데이터로 공격을 하는 것이 보통이다. 그런데 어떤 네트워크의 라우터에서는 source 주소가 위장된 패킷은 바깥으로 나가지 못하도록 설정되어 있는 경우가 있다.

이때는 agent 데몬은 자신의 핸들러를 찾는 것 외에도, 현재 agent가 속해있는 네트워크에서 위조된 source address를 가지는 패킷이 네트워크 바깥으로 나갈 수 있는지를 테스트한다. Agent 데몬은 IP 헤더 부분에 위조된 source IP 주소, ICMP ID필드에 "666", 그리고 데이터 필드에 agent의 IP 주소를 포함하는 ICMP ECHO request 패킷을 master시스템의 핸들러에게 보낸다.

보통 ICMP패킷의 헤더 부분에는 ICMP메시지의 종류를 구분하기 위해 type영역이 있는데 agent 데몬이 master로 보내는 source 주소가 위조된 테스트 패킷에는 이 type값이 7이 들어가게 되는 것이 특징이다.

master시스템의 핸들러가 이 패킷을 수신하게 되면 핸들러는 다시 ICMP ECHO_REPLY 패킷으로 테스트가 성공적이라는 답신을 보내게 된다. 이 패킷의 ID 필드에는 1000, 데이터 필드에는 "spoofworks"값을 포함하여 agent 데몬이 보낸 테스트 패킷의 데이터필드에 들어 있는 진짜 agent IP 주소로 보내어 진다. agent 데몬이 이 답신을 받게 되면 agent 데몬은

spoof_level을 0으로 설정하는데, 이것은 IP주소를 표현하는 32비트 모두를 위조 가능한 레벨이다.

4. 확장된 IDEF 스키마

1) Stacheldraht툴의 DDoS공격 특징

이제까지 stacheldraht의 동작을 살펴보았듯이 master시스템의 핸들러와 agent 데몬은 서로 어떤 특정한 통신을 하기 위하여 암호화되지 않은 상태의 ICMP 패킷을 주고받는다. 그 경우 ICMP패킷은 다음과 같이 특정필드에 특정 값을 갖는다.

ID	data	type	source	destination
666	skillz	0	agent	master
667	ficken	0	master	agent
666	위조된 source IP주소	7	agent	master
1000	spoofworks	0	master	agent

표 10 : Stacheldraht툴 공격시 패킷데이타

이것은 stacheldraht툴이 공격을 하기 위한 전처리 작업이며, 그 툴이 가지는 특성이다. 따라서, 네트워크 ICMP패킷을 모니터링함으로써 stacheldraht툴이 깔려있는 master와 agent를 찾아 낼 수 있는 방법이 된다.

2) 추가적인 자료형식

이전에 IDEF에서 제안한 데이터모델에 다음과 같이 ICMP 패킷의 ID, 데이터필드, type을 고려한 stacheldraht스키마를 첨가하였다. 이를 이용해 stacheldraht툴을 이용한 공격에 대한 새로운 정보들을 데이터베이스화하고, stacheldraht툴을 이용해 공격이 발생하는 경우 이를 탐지할 수 있다.

ICMP packet sequence #	type	code	ID	data

표 11 : 제안하는 데이터 스키마

IDEF에 본 연구에서 제안한 스키마를 통합하여 전체적으로 이루어지는 작업은 다음과 같다.

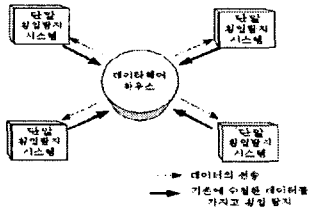

그림 4 : 제안한 네트워크 모델

네트워크 상에 한 개의 데이터웨어하우스와 한 개 이상의 호스트가 연결되어 있다. 그리고, 이미 각각의 호스트에는 임의의 IDS프로그램이 설치되어 있다. 이 IDS제품들도 misuse나 abnormal한 데이터를 탐지해 내기 위한 각자의 데이터베이스시스템을 가지고 있다. 이런 상황에서, 어떤 한 호스트에 stacheldraht들이 백도어로 설치되어 있고(master혹은 agent로써), 그것이 attack을 하기 위해서 앞에서 이야기한 작업들을 시도한다면 IDS자체에서 설정된 룰에 의해 이상여부를 판단 할 수 있다. 그리고, 그 ICMP패킷과 관련해서 발생한 정보들을 데이터웨어하우스로 보내게 된다. 데이터웨어하우스에서는 IDEF에서 제안한 모델과 본 논문에서 제안한 모델을 갖고 있으며 그 패킷과 관련된 정보를 새로 저장하게 된다. 이렇게 데이터은 지속적으로 데이터웨어하우스에 수집 관리될 것이고, 이것은 다른 데이터들과 병행하여 침입이나 이상행위를 탐지하게 된다.

5. 결론

본 논문에서는 통합 침입탐지 시스템에서 상위 시스템에서 수집된 경보 자료를 관계형 데이터베이스에서 관리 할 수 있도록, IDEF모델을 관계형 데이타베이스 스키마로 변환하는 방법을 설명하였다. 그리고, 통합 침입탐지시스템에서 추가적으로 필요한 자료에 의거하여 IDEF모델을 확장하기 위한 시도로서, DDoS공격 탐지에 필요한 자료 형태에 관해 고찰하고 확장된 스키마를 제안하였다.

향후에는 다른 DDoS툴을 이용한 공격들에도 IDEF을 확장할 수 있는 방법을 더 연구해야 할 것이다. 그리고, 각각의 DDoS툴에 대한 정보를 통합하여 하나의 스키마로 만들어 DDoS공격을 탐지하는 방법에 대해 연구할 예정이다.

참고문헌

[1] IDEF모델 http://www.oasis-open.org/cover/draft-ietf-idwg-data-model-03.txt

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

[3] 오라클 테크널러지 네트워크 http://otn.oracle.co.kr

[4] EMERALD : Event Monitoring Enabling Response to Anomalous Live Disturbance, 1997, SRI

[5] Stacheldraht분석 http://staff.washington.edu/dittrich/misc/stacheldracht.analysis.txt