프로세스 상태 모니터링을 통한 효율적인 침입탐지시스템

남중구[0] 임재걸
동국대학교 전자계산학과[0] 컴퓨터학과
{junggu, yim}@wonhyo.dongguk.ac.kr

An Efficient Intrusion Detection System By Process State Monitoring

Joong-Goo Nam[0] Jae-Geol Yim
Dept. of Computer Science, Dongguk University

요 약

침입탐지의 종류를 탐지 방법 측면에서 구분해보면 크게 이상탐지와 오용탐지로 나뉘어진다. 침입탐지의 주된 목적은 탐지오류를 줄이고 정확한 침입을 판가름하는데 있다. 그러나 기존의 이상탐지와 오용탐지 기법은 그 방법론상에 이미 판단오류 가능성을 내포하고 있다. 이상탐지는 정상적인 사용에 대한 템플릿을 기초로 하므로 불규칙적인 사용에 대처할 수 없고, 오용탐지는 침입 시나리오라는 템플릿에 기초하므로 알려지지 않은 침입에 무방비 상태인 문제가 있다. 침입의 주요 목적은 관리자의 권한을 얻는 것이며, 그 상태에서 쉘을 얻은 후 원하는 바를 행하는 데 있을 것이다. 그러므로 그 상태를 얻으려는 프로세스와 추이와 결과를 모니터링하여 대처하면 호스트기반 침입의 근본적인 해결책이 될 수 있다. 그러므로 본 연구에서는 프로세스의 상태를 모니터링함으로써 컴퓨터시스템의 침입을 탐지하는 새로운 기술에 대해 제안하고 설명한다. 프로세스의 상태는 일반상태, 특권상태, 관리자상태 등으로 구분되며, 시스템에 의해 부여된 실사용자ID, 유효ID, 실그룹ID, 유효그룹ID를 점검함으로써 이루어진다. 본 연구에서 모니터링에는 BSM을 사용하며, 호스트기반에서 사용한 프로세스의 상태 모니터링에 의한 침입탐지시스템 구현한다.

1. 서 론

침입탐지 시스템은 불법적인 사용, 요용, 또는 불법적인 사용자나 외부 침입자에 의한 컴퓨터 시스템을 남용하는 침입을 알아내고자 하는 시스템이다[1]. 이러한 시스템은 단일 컴퓨터나 네트워크로 연결된 여러 시스템을 감독하며 감사 기록, 시스템 테이블, 네트워크 부하(traffic) 기록의 자원으로부터 사용자 행위에 대한 정보를 분석하여 수행된다.[2]

침입탐지의 방법 측면에서 침입탐지시스템(IDS: Intrusion Detection System)을 분류해보면 하나는 컴퓨터 자원의 비정상적인 행위나 사용을 탐지하는 이상탐지(anomaly detection)이고, 다른 하나는 시스템이나 응용 프로그램의 약점을 통하여 시스템에 침입할 수 있는 잘 정의된 공격을 탐지하는 오용탐지(misuse detection)이다[3]. 여기서 침입탐지에 대한 현재 기술적 실제적 논점은 실제로는 침입이 아닌데 침입으로 판정하는 경우(false positives)와 실제로는 침입인데 탐지하지 못하는 경우(false negatives)를 어떻게 최소화하며, 적절히 처리하는가에 있다[4].

또한 침입탐지를 침입의 침입경로면에서 구분해 보면 호스트기반과 네트웍 기반으로 나눌 수 있다. 호스트기반이란 불법적인 사용자 로그인을 포함하여 일반 계정상태에서 관리자 권한을 얻으려는 모든 시도를 포함하는 침입이며, 네트웍기반 침입이란 스누핑, 스니핑, 서비스거부 공격 등을 포함하여 네트웍 경로를 통한 자원을 탈취 및 사용정지를 목적으로 하는 것이다[7]. 본 연구에서는 솔라리스 7을 탑재한 유닉스 호스트 기반에 대한 것으로 네트웍기반 침입은 고려하지 않는다.

2. 연구배경

본 절에서는 기존연구와 배경지식에 대해 언급한다.

2.1 이상탐지와 오용탐지

이상탐지와 오용탐지를 초기에 정의하는 것이 무엇인가에 따라 구분해보면, 이상탐지는 정상적인 사용을 정의하는 것이고, 오용탐지는 침입행위 자체를 정의하는 것이라 할 수 있다.

먼저 이상탐지는 정상적인 사용을 먼저 정의하고, 일반 사용자가 그 기준에 얼마나 부합된 사용을 하는지를 검사하는 방법으로, 항상 비슷한 사용을 하는 사용자들로 구성된 환경에서는 적용할 수 있으나 변칙적인 사용자를 많이 가진 학교나 벤처기업 같은 곳에서는 적용하기 어려운 문제가 있다. 학교의 경우 과제가 많은 시기에 사용자가 몰릴 수 있고, 과제의 성격이나 진도에 따라 사용양상이 판이하게 달라지기 때문이다. 벤처기업의 경우도 심야작업이 연일 계속되기도 하고 업무가 없어 사용을 하지 않는 시기도 있으므로 부적당하다. 또한 통계적 기법에 기반하므로 구조적으로 침입이 아닌 것을 침입이라 판단(false negatives)하는 가능성을 내포하고 있는 문제를 안고 있다[9].

이상탐지는 침입행위 자체를 정의해 두고 일반 사용자의 이벤트가 그 정의에 부합되는지를 검사하여 탐지하는 방법이다. 이 방법은 이미 정의된 방법에 대해서는 정확성이 뛰어나지만, 급변하고 다양화되는 침입기술을 따라가기 힘든 문제가 있다. 즉, 새로운 침입기술이 등장할 때마다 침입 시나리오 데이터베이스를 갱신해야 하는 문제가 있고 알려지지 않은 방법으로 접근해 올 때 무방비 상태인 문제가 있어서 이 역시 구조적으로 침입인데 침입이 아닌 것으로 판명(false negative)하는 문제를 안고 있다.

1999년 Nittida Nuansri 등에 의해 프로세스의 상태전이 분석과 침입탐지에의 적용에 관한 연구가 있었다[5]이 연구에서는 사용자 프로세스를 모니터링하기 위해 BSD의 ktrace()라

는 시스템콜을 수정 사용하였다. 본 연구에서는 이 연구를 기반으로 ktrace 대신 SUN solaris의 내장 모듈인 BSM을 재적용 및 보완하여, 기존의 이상탐지와 오용탐지의 문제점을 극복하기 위해 사용자의 식별자와 권한 상태를 모니터링하여 허락받지 않은 권한상태에 있을 시 침입으로 간주하는 방법을 사용한다

2.2 Unix의 사용자 식별과 권한

유닉스시스템에서 사용자는 프로세스 실행 시에 사용자식별자(UID)와 그룹 식별자(GID)에 의해 식별된다. 또한 유닉스에는 프로세스 실행 시에 프로세스가 어떤 권한을 가지는지를 나타내는 특수한 식별자를 effective UID(EUID)라 한다. 일반적으로는 UID와 EUID는 동일하며 사용자에 의해 실행된 프로그램 프로세스의 EUID는 그 사용자의 UID로 설정되게 된다. 그러나 프로그램에 setuid비트라는 특별한 비트가 설정되어 있으면 사용자에 의해 실행된 프로세스의 EUID는 그 사용자의 UID가 아니라 프로그램파일의 UID로 설정되게 된다. 이와 같은 방법으로 setgid비트도 존재한다[6].

2.3 BSM

BSM(Basic Security Module)은 Solaris 2.3부터 포함된 보안모듈로, 시스템콜 레벨 모니터링을 가능하게 함으로써 유닉스 보안을 C2등급으로 상승시켜준다. BSM이 수행되는 시스템에서는 로그인 시 audit ID가 사용자의 프로세스에 할당되며 로그아웃시까지 사용자 레벨 및 시스템콜 레벨의 감사기록을 제공한다. 이 때 su나 백도어를 통해 ID가 바뀌더라도 동일 ID로 추적하게 된다. BSM은 보안관련 이벤트 모니터링은 물론 audit trail 분석에 의한 오용탐지 및 비 인가된 행위의 탐지도 가능하게 한다[8][10].

[표 1]은 BSM 감사레코드의 전형적인 구조를 보인 것이다.

[표 1] BSM audit record의 전형적인 구조

- head token : 레코드 전체에 대한 헤드정보
  token ID(1byte) 레코드길이(4byte) 토큰버전(1byte) event ID(2byte) ID modified(2byte) date and time(8byte)
- arg token : 시스템콜의 인자정보
  token ID(1byte) argement#(1byte) argument value(4byte) text length(2byte) text(nbyte)
- subject token : 프로세스 상태정보
  token ID(1byte) audit ID(4byte) user ID value(4byte) group ID(2byte) real user ID(4byte) real group ID(4) process ID(4byte) session ID (4byte) terminal ID)deviceED, machine ID(4)
- return token : 시스템콜의 리턴상태정보
  token ID(1byte) process error(1) process value(4byte)

2.4 프로세스의 상태정의

프로세스의 각 식별자는 일반권한, 특수권한, 그 외 권한으로 구분할 수 있다. 편의를 위해 실행자와 같은 권한 식별자를 uid, 특수권한 식별자를 sid, 그 외의 다른 사용자 권한을 oid라 칭하기로 하고 그룹에 대해서도 gid, sgid, ogid라 하기로 한다. 이를 바탕으로 실행된 프로세스가 가지는 UID, EUID, GID, EGID의 네 쌍의 식별자로 구분되는 4가지 상태를 정의한다.

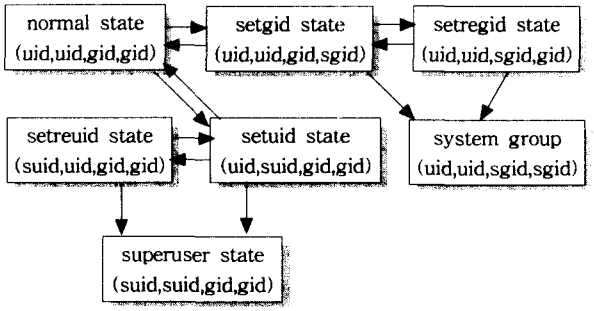
정의 1: 정상상태
상태의 네 가지 프로세스 식별자가 (uid, uid, gid, gid)이면 정상상태라 한다.

정의 2: 특권상태
프로세스의 상태값 중의 하나가 특권을 가진 식별자일 때 특권상태라 한다.

정의 3: 관리자상태 및 시스템 그룹 상태
real UID와 effective UID가 모두 특권사용자 ID일 때를 관리자 상태라 하고
GID와 EGID가 모두 특권 그룹 ID일 때 시스템 그룹 상태라 한다.

정의 4: 그 외 사용자 상태
다음 중 하나일 때를 말한다.
(1) RUID와 EUID가 모두 다른 사용자의 UID(oid)로 변경된 경우
(2) GID와 EGID가 모두 다른 사용자의 GID(ogid)로 변경된 경우.

[그림 1]은 프로세스의 상태 전이 다이어그램을 보인 것이다.



[그림 1] 프로세스 상태 전이 다이어그램

2.5 IDS를 위한 규칙

특정 사이트의 보안정책에 따라 본 규칙은 달라질 수 있으며 본 연구에서는 다음을 그 기본 정책으로 한다.

Rule 0: 오직 setreuid()와 setregid() 시스템 콜만이 UID나 GID를 각각 변경할 수 있다.
그 외의 어떠한 시스템 콜이나 프로그램이 RUID나 RGID를 변경하려 하면 침입으로 간주한다.

Rule 1: 특권상태에서는 어떠한 execve() 콜도 허용하지 않는다.
execve()에 의해 생성된 어떤 자식 프로세스도 부모의 권한을 상속하므로 특권상태에서 execve()가 허용되어서는 안된다.

Rule 2: 특권상태의 프로세스는 setuid/setgid 프로세스를 생성하는 것을 허용하지 않는다.
오직 superuser에게만 이것을 허용한다.

Rule 3: 프로세스는 시스템 프로그램을 수정하도록 허용되지 않는다.

Rule 4: 관리자상태가 아닌 어떤 프로세스도 새로운 계정을 생성하려는 시도를 할 경우 침입으로 간주한다.

3. 실험

본 실험에서는 일반사용자가 대입에 의한 관리자 암호추측이나 스누핑에 의해 관리자의 암호를 알아냈다고 가정하고 su명령을 사용해 관리자권한을 얻은 것에 대해 탐지하는 결과를 보이기로 한다.
[표 2]와 [표 3]은 ids라는 사용자가 su를 사용해 root쉘을 얻으려는 과정의 audit레코드를 보인 것으로(좌측의 번호는 편의를 위해 붙인 것) 64번에서 su를 실행하여 73번에서 c쉘을 수행하여 성공적으로 쉘을 얻었으나 모니터링에 의해 Alert가 되고 프로세스가 제거되는 것을 보여주는 것이다.

[표 2] su에 의한 root권한 획득을 보이는 audit레코드

```
...
62  return,success,0
63  header,104,2,execve(2),,Sep 20 16:16:43 2001, + 379983000
    msec
64  path,/usr/bin/su
65  attribute,104555,root,sys,8388624,56971,0
66  subject,ids,ids,user,ids,user,639,624,24 1
    cspc197.dongguk.ac.kr
67  return,success,0
68  header,86,2,su,,Sep 20 16:16:46 2001, + 219981500 msec
69  subject,ids,root,user,ids,user,639,624,24 1
    cspc197.dongguk.ac.kr
70  text,success for user root
71  return,success,0
72  header,105,2,execve(2),,Sep 20 16:16:46 2001, + 259983500
    msec
73  path,/usr/bin/csh
74  attribute,100555,bin,bin,8388624,56860,0
75  subject,ids,root,other,root,other,639,624,24 1
    cspc197.dongguk.ac.kr
76  return,success,0
...
```

[표 3] 모니터링 및 대처 결과

```
...
System Call : execve(2) ,Sun Sep 23 16:16:43 2001
/usr/bin/su 639 ids (ids,user,ids,user) Success
System Call : su ,Sun Sep 23 16:16:43 2001
/usr/bin/su 639 ids (root,user,ids,user) Success
Alert : may be attack
System Call : execve(2) ,Sun Sep 23 16:16:43 2001
/usr/bin/csh 639 ids (root,other,root,other) Success
Alert : pid 639 killed
...
```

4. 결론

본 연구에서는 오용탐지와 이상탐지가 가지는 근원적인 문제를 극복하는 침입탐지시스템의 방법론을 제시하고 구현하였다. 시스템은 간결하고 오동작없이 정확하게 작동하며 호스트기반에서 root쉘을 얻으려는 침입에는 적합하다. 그러나 근래의 침입방법은 매우 다양화 되어가고 있어 다음 연구에서는 불법적인 자원에 대한 접근탐지 등의 선처리가 추가되어야 더 지능적으로 대처할 수 있을 것이다.

참고문헌

[1] D. E. Denning, "An Intrusion-Detection Model." IEEE Trans. on Software Engineering. No.2. Feb., 1987

[2] 김민수, 은유진, 노봉남, "UNIX 환경에서 퍼지 Petri-net을 이용한 호스트 기반 침입 탐지 모듈 설계, 한국정보처리학회 논문지 제 6권 제7호(99.7)

[3] T. F. Lunt. "A Survery of Intrusion Detection Technique." Computer & Security. Vol12. No.4, Jun. 1993

[4] Sandia National Lab., "Intrusion Detection and Response." National Info-Sec Technical Baseline. http://doe-is.llnl.gov/nitb/docs/nitb.htm, Oct., 1996

[5] Nittida Nuansri, Samar Singh, Tharam S. Dillon, "A Process State-Transition Analysis and its Application to Intrusion", IEEE

[6] W. Richard Stevens, Advanced Programming in the UNIX Environment, Addison Wesley 1992 p.77

[7] 김수형, 강명호, 조형재, 송주석, 안전하고 효율적인 침입자 역추적 시스템, 정보과학회 논문지(A) 제 25권 제 10호

[8] Koral Ilgun, USTAT: A Real-time Intrusion Detection System for UNIX, IEEE Computer Society Symposium on Research in Security and Provacy. May, 1993 proceed., pp. 16-28

[9] Nicholas J. Puketza, Kui Zhang, Mandy Chung, A Methodology for Testing Intrusion Detection Systems, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 22, NO. 10, OCTOBER 1996

[10] Sun Microsystems, Inc, SunSHIELD Basic Security Module Guide, Part No:805-2635-10 Oct 1998