KIPS Tr. Comp. and Comm. Sys.
Vol.3, No.11 pp.427~432 pISSN: 2287-5891

http://dx.doi.org/10.3745/KTCCS.2014.3.11.427

A Study for Task Detection Acquiring Abnormal Permission in Linux

Kim Won-iL[†] · Yoo Sang-Hyun[††] · Kwak Ju-Hyun[††] · Lee Chang-Hoon[†††]

ABSTRACT

The Purpose of local system attacks is to acquire administrator's(root) privilege shell through the execution of the malicious program or change the flow of the program. This acquiring shell through attack is still valid approach method and it is difficult to cope with improving each of vulnerability because the attacker can select various forms of attack. Linux allocate a set of credentials when login, in order to manage user permissions. Credentials were issued and managed by the kernel directly, and also the kernel ensures that any change cannot be occurred outside of kernel. But, user's credentials that acquired root privilege through system attacks occurs a phenomenon that does not remain consistent. In this paper we propose a security module to detect a security threats that may cause to users and tasks by analysis user task execution and inconsistency credentials.

Keywords : Credentials, Security Module, Privilege Escalation, IDPS

리눅스의 비정상 권한 획득 태스크의 탐지방법 연구

김 원 일[†] · 유 상 현[††] · 곽 주 현[††] · 이 창 훈[†††]

요 약

로컬 시스템에 대한 공격은 프로그램의 흐름을 변경하거나, 악의적인 프로그램의 실행을 통해 관리자 권한의 쉘을 획득하는 것을 목적으로 한다. 공격을 통해 쉘을 실행하는 방법은 현재까지도 유효한 방법이며, 공격자는 다양한 형태로 공격을 수행하기 때문에 각각의 취약점을 개선하는 것으로는 대처가 어렵다. 리눅스는 사용자 권한 관리를 위해 로그인 시에 커널이 발급하는 권한의 집합인 자격증명을 할당한다. 자격증명은 커널이 직접 발급 및 관리하고, 커널 외부에서 변경되지 않을 것을 보장한다. 그러나 시스템 공격을 수행하여 관리자 권한을 획득한 사용자는 자격증명 일관성이 유지되지 않는 현상이 발생한다. 본 논문에서는 이러한 자격증명이 불일치한 사용자의 태스크 실행 요청을 분석하여 보안 위협이 발생할 수 있는 사용자와 태스크를 탐지하는 보안 모듈을 제안한다.

키워드 : 자격증명, 보안 모듈, 권한 상승, 침입 탐지 및 예방 시스템

1. 서 론

네트워크와 하드웨어의 급속한 발전은 컴퓨터 시스템의 이동성과 편리성을 급격히 발전시켰다. 그에 반해 운영체제의 발전은 더딘 편이며, 대부분의 발전 방향은 보안적인 측면보다는 기능성에 치중하였기 때문에 많은 취약점들을 발생시켰다[1]. 이로 인해 다양한 형태의 보안 시스템들과 기술 및 프로그램들의 발전이 이루어졌고, 대표적으로 네트워크와 시스템에 대한 공격들을 탐지하는 형태인 침입 탐지 및 예방 시스템(Intrusion Detection and Prevention Systems: IDPS)이 발전하였고, 이를 발전시켜 NIST에서는 탐지 및 예방을 위한 가이드라인을 제안하였다[2]. 공격자들의 침입 방법은 매우 다양하고 또한 정형화되어있지 않기 때문에 침입에 대한 각각의 대응은 높은 효과를 거두기 어렵고 또한 우회할 수 있는 방법들이 존재한다[3, 4, 5]. 대부분의 침입 탐지는 네트워크에 관련된 탐지 연구가 이루어졌는데, 네트워크를 통한 침입 방법이 다양하고 관련 프로그램의 급증으로 보다 많은 수의 취약점이 존재하기 때문이다. 그러나 네트워크 침입은 결국 로컬 시스템 자원에 대한 접근이나 정보의 취득을 위해 보다 높은 권한의 획득을 필요로 한다[6].

정상적인 절차를 통해 인증된 사용자는 자원 접근을 위해 커널로부터 권한을 할당받는데, 이를 관리하기 위해 자격증명이라는 권한 집합체를 이용한다. 한번 할당된 자격증명은 외부에서 임의로 변경이 불가능하며, 권한의 일관성 유지를 위해 동일한 값을 갖는다. 현재 사용자보다 높은 권한을 획득하는 시스템 취약점 공격을 권항 상승 공격이라 하며, 정

† 준 회 원 : 유한대학 컴퓨터정보과 강의전담교원
†† 준 회 원 : (주)아이지코 연구원
††† 종신회원 : 건국대학교 컴퓨터공학과 교수
Manuscript Received : July 29, 2014
First Revision : September 18, 2014; Second Revision : October 20, 2014
Accepted : October 20, 2014
* Corresponding Author : Lee Chang-Hoon(chlee@konkuk.ac.kr)

상적인 권한 획득 방법인 로그인이나, “su”를 거치지 않고, 관리자 권한의 쉘을 획득하는 것을 목표로 한다. ROP[7], return-into-libc[8]와 같은 공격은 권한 상승 공격을 수행하는 대표적인 예로 특정 라이브러리의 주소를 변조하거나 공격 코드의 분산과 재사용을 통해 일시적 권한 상승이 발생한 시점에 관리자 권한의 쉘을 실행한다. 일시적 권한 상승은 사용자의 모든 id값을 관리자로 변경하지만, 그룹 리스트의 값을 변경하지 않으므로 이전 권한 정포가 남은 채로 실행된다. 이 과정에서 실행되는 쉘은 이전 권한 정보가 남은 자격증명을 포함하여 실행되어 권한의 불일치가 발생한다. 따라서 권한 상승 공격을 통해 비정상적으로 획득된 관리자 권한은 관리자보다 낮은 권한 정보가 자격증명의 그룹 리스트에 남게 된다.

본 논문에서는 자격증명의 권한과 그룹이 불일치하면서 현재 권한보다 낮은 권한이 존재하는 자격증명을 비정상 권한의 획득으로 규정하고, 이를 탐지하는 방법을 제안한다. 또한 공격자가 자격증명을 새로 발급받아 완전한 자격증명을 획득하여 실행하는 태스크의 탐지를 통해 보안 문제 발생 소지가 있는 태스크들을 격리하고, 이를 중지할 수 있는 보안 모듈을 설계하고 구현하였다.

2. IDPS

침입 탐지와 예방 시스템은 크게 네트워크와 시스템에 대해 허용되지 않은 사용자의 시스템 접근을 막고, 앞으로의 침입을 막기 위한 방법을 통칭한다. 대부분의 침입 탐지는 공격이 발생한 패턴을 학습하거나 보안 정책에 반하는 행위들을 탐지하는 형태로 구성된다. 여기에 침입 또는 침입에 의한 행위들의 로깅을 수행하며, 침입이 발생하였다면 이를 제지하는 일련의 작업을 수행한다. 이러한 침입 탐지와 예방 시스템은 다음과 같이 4가지 분류로 크게 나뉜다[2].

- 네트워크 기반
- 무선 랜 기반
- 네트워크 행위 분석 기반
- 호스트 기반

네트워크, 무선 랜, 네트워크 행위 분석 기반 침입 탐지와 예방 시스템은 패킷에 관련된 정보들을 기반으로 이루어지며 보통 NIDPS로 명명된다. 호스트 기반 침입 탐지(HIDPS)는 한 시스템에서 발생하는 위험분석 및 의심 행위의 수집과 이들의 특징을 분석하여 시스템에 대한 침입을 탐지하거나 예방하는 형태로 동작한다. 이러한 침입 탐지 및 예방 시스템은 안정성과 유효한 정보의 수집 및 로깅의 일관성을 유지하면서 탐지와 예방이 가능해야 한다. 또한 시스템에 영향을 최소화하면서 탐지가 가능해야 하며, 침입이 발생한 경우 관리자에게 알림과 시스템 구현 및 유지 보수와 관리가 가능해야 하는 요소들을 만족할 것을 요구하고 있다[2].

호스트 기반 침입 탐지 시스템은 파일 시스템 모니터링, 로그 파일 분석, 연결 분석, 커널 기반 탐지로 분류된다[9]. 보통 호스트 기반 침입 탐지 시스템은 전체 시스템을 모니터링 해야 하기 때문에 항상 실행 중이어야 하며, 대부분 시스템에 설치되는 형태로 구성된다. 기본적으로 호스트 기반 침입 탐지 시스템은 시스템호출이 발생하는 순서의 정규화 및 학습을 통해 공격자의 시스템호출 순서가 갖는 일련의 호출 패턴을 갖는 경우 이를 침입으로 판단하고 탐지하는 형태로 구성되었다. 최근에는 공격이 가능한 시스템 호출 순서를 신경망을 통해 학습하여 침입을 탐지하는 연구가 이루어지고 있다[10]. 이러한 시스템 호출에 기반을 둔 탐지방법은 학습을 위한 초기 비용이 높고, 학습된 정보를 통해서만 침입을 판단하기 때문에 동적으로 변화하는 공격에 대한 대비가 어렵다.

3. 비정상 권한 획득 태스크 탐지

3.1 자격증명 정보의 확인

자격증명(Credentials)은 사용자가 시스템 자원에 접근하기 위해 필요로 하는 모든 권한을 포함하며 “/include/linux/cred.h”의 cred 구조체에 정의되어있다. 자격증명은 사용자가 인증된 이후 단 한 번만 할당되며, 커널이 사용자의 요청을 처리하기 위한 특별한 경우를 제외하고는 커널 외부에서 임의적인 변경이 불가능하다. 또한 RCU(Read Copy Update[11]) 나 특별한 잠금을 수행하지 않더라도 일관성을 유지하도록 커널이 보장하고 있다[12]. 사용자의 요청을 처리하는 대표적인 경우는 시스템호출(system call)로, 시스템호출 시에 변경된 자격증명은 처리가 정상적으로 수행되면 원래 권한으로 되돌아간다. 따라서 자격증명 정보가 인증 이후에 변경되었다는 것은 임의 또는 강제적인 변경이나, 커널이 의도하지 않은 권한 변경이 발생하였음을 의미한다.

인증된 사용자의 권한 정보를 확인하는 방법은 간단히 “id” 명령어를 이용한다. “id” 명령어는 현재 터미널에 연결되어있는 사용자의 권한과 소속된 그룹 정보들을 지정된 형태로 출력한다. 또한 getuid(), getgid(), getgroups() 등의 시스템호출을 이용하여 현재 사용자 세션에 연결된 자격증명 정보들을 확인할 수 있다. 이들을 이용하여 사용자 자격증명을 출력한 결과는 Fig. 1과 같다. 즉, 시스템호출을 이용한 결과와 “id” 명령어를 이용한 출력 결과에서 (e)uid, (e)gid와 groups에 첫 번째로 나타난 그룹 정보가 동일하며, “id” 명령어에서 출력된 그룹의 개수와 그룹 리스트가 동일함을 알 수 있다.

정상 인증 과정을 거쳐 “su” 명령으로 관리자 권한을 할당받으면 Fig. 2A와 같은 형태의 출력 결과가 나타난다. Fig. 1과 같이 각 id값과 그룹의 개수가 동일하게 출력되는 것을 확인할 수 있다. Fig. 2B는 최근 권한 상승 공격의 대표적인 ROP 공격을 통해 관리자 쉘을 획득하고 자격증명을 출력한 결과이다. ROP는 전체 공격 코드를 가젯이라는 작은 단위로 나누어 프로그램에 배치하고, 이를 연속적으로 실

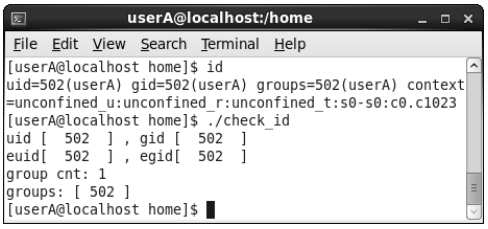


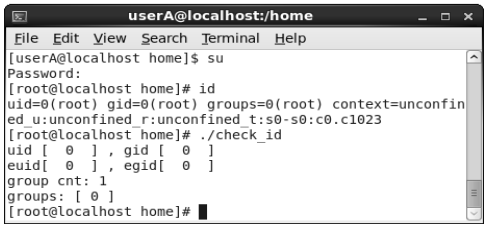
Fig. 1. Example of Credentials

행할 수 있도록 스택을 조작하는 공격 방법이다[7]. 먼저 "id" 명령어의 groups의 출력에서 관리자 이외의 사용자 정보가 출력되는데, 이는 공격을 수행한 사용자인 userA임을 알 수 있다. 또한 "id" 명령으로 출력된 groups는 2개인 데 반해, "check_id" 프로그램의 출력에서 "group cnt"가 1로, "groups" 또한 관리자가 아닌 userA의 아이디 값임을 알 수 있다. 이러한 현상은 groups 리스트를 저장하는 배열의 첫 번째 인덱스가 로그인을 통해 세션이 연결된 사용자의 아이디로 고정되어있는 특징에 기인한다[12].

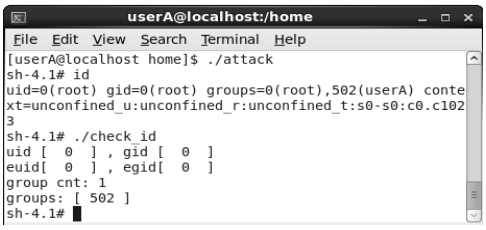
따라서 비정상적인 권한 상승이 발생한 자격증명은 일관된 정보를 유지할 수 없을 뿐만 아니라 현재 사용자보다 낮은 권한을 갖는 자격증명을 groups에 포함하게 된다. 본 논문에서는 이렇게 일관성이 유지되지 않으면서 권한 상승이 발생한 것을 비정상 권한 획득으로 규정하고 이를 탐지한다.

3.2 리눅스의 태스크와 자격증명

리눅스의 모든 태스크의 실행은 부모 태스크의 복사를 통해 이루어진다. fork()/execve()은 태스크의 복사를 통해 새로운 태스크를 실행하는 가장 잘 알려진 시스템 호출이다. 태스크의 실행을 위한 복사는 메모리의 복사와 동시에 자격증명의 복사도 수반한다[12]. 즉, 부모와 자식 태스크의 자격증명은 항상 동일한 정보를 유지하여 권한 문제가 발생하지 않도록 구성되며, 임의적인 변



(A) Credentials for root with "su" command

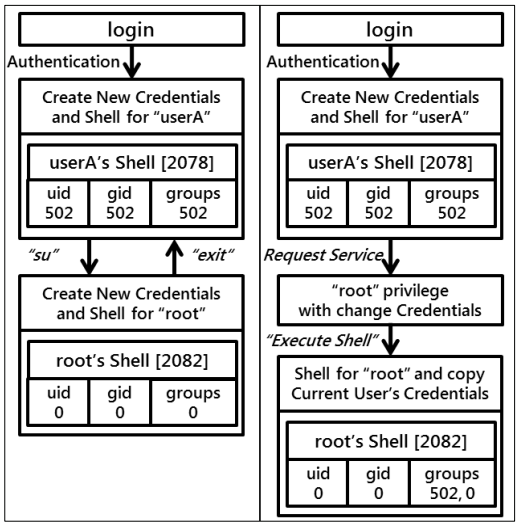


(B) Credentials for root with ROP Attack

Fig. 2. Difference of Credentials

경이 불가능하다. 따라서 태스크의 자격증명에 Fig. 2B와 같은 현상이 발생한다면 비정상 권한 상승 태스크로 볼 수 있다.

인증된 사용자가 자격증명을 완벽하게 변경할 수 있는 방법은 "su" 명령어를 이용하는 것이다. 일반적인 태스크의 실행과 달리 "su"는 현재 사용자의 자격증명을 복사하지 않고, 인증이 완료된 시점에 새로운 세션을 열고 자격증명을 새로 생성한다. 따라서 "su" 명령을 실행한 쉘과 "su"에 의해 실행된 쉘은 각기 다른 자격증명이 할당된다[13].



(A) (B)

Fig. 3. (A) "su" Command Flow
(B) Abnormal Credential Acquiring Flow

Fig. 3A는 인증된 사용자가 "su" 명령어의 실행을 통해 자격증명을 변경하여, 새로운 쉘을 실행한 경우의 자격증명을 나타낸다. Fig. 3B는 공격을 통해 자격증명의 불일치가 발생한 태스크의 자격증명 흐름을 나타낸다. 시스템호출을 통한 일시적 권한 상승의 발생은 문맥 교환이 발생하지 않고, 일시적으로 권한만 변경되어 처리된다[14]. 따라서 일시적 권한 상승이 발생한 시점에 태스크, 대부분 쉘을 실행하는 공격은 권한 상승 발생 이전의 그룹 정보가 남아 자격증명의 불일치를 발생시키지만 부모 pid는 변경되지 않는 특징을 갖는다. 즉, 공격에 의한 태스크 실행은 부모 태스크에 대한 정보를 유지하면서 자격증명이 불일치하는 특징을 갖는다.

본 논문에서는 공격을 통해 최종적으로 관리자 권한의 쉘을 실행시킨 프로그램이 갖는 자격증명 정보의 불일치와 실행된 쉘의 pid 정보를 토대로 권한 상승이 발생한 자격증명을 갖는 태스크를 탐지하는 모듈을 제안한다.

4. 탐지모듈의 설계 및 구현

4.1 제안 모듈의 구성

공격자에 의해 관리자 권한의 쉘이 실행되는 공격들은 3절에서 살펴본 바와 같은 자격증명의 불일치를 이용하여 공격자 또는 태스크를 탐지하는 데 활용한다. 제안 모듈은 시

스템에 발생하는 부하를 최소화하고, 시스템의 동작을 방해하지 않으면서 커널의 수정 없이 정보를 수집하여, 자격증명이 변경된 태스크의 탐지를 목표로 한다. 이러한 커널 기반 정보 수집 방식은 후킹을 이용하는 것이 대표적이다. 즉 시스템에 자원을 요청하는, 특히 파일 시스템에 접근하거나 새로운 태스크를 실행하는 시스템호출 시점에 검사하는 것이 효율적이다. 이 방식은 이미 LSM(Linux Security Module)에서 그 성능의 우수함이 증명되었다. LSM은 Fig. 4와 같이 요청 처리 후, 결과를 반환하기 전인 DAC(임의적 접근 제어)의 수행 후에 동작한다[15]. 즉, 요청에 대한 작업이 모두 수행된 이후에 접근 허용 여부를 결정하기 때문에 접근이 거부된다면 이전 상태로 되돌리기 위한 비용이 발생한다. 그러나 제안된 모듈이 정보를 수집하고 처리하는 과정은 LSM과 달리 시스템 콜이 호출된 시점에 처리 가능하다. 따라서 이후 커널의 작업 자체의 수행 여부를 결정할 수 있기 때문에 시스템에 미치는 영향이 적다. 또한 자격증명과 태스크에 대한 접근은 current 매크로를 통해 직접 접근할 수 있기 때문에 추가적인 처리가 필요 없다.

태스크의 실행은 대부분 execve()를 이용하고, 자원에 대한 접근은 파일로 추상화되어 처리되기 때문에, 자원에 대한 접근은 결국 파일처리를 위한 시스템호출을 이용하게 된다. 따라서 태스크를 실행하는 execve()와 파일처리를 위한 open(), read(), write(), close() 시스템호출을 후킹하여 자격증명을 확인하고 수집한다. 이러한 시스템호출 후킹에서 자격증명이 불일치하는 태스크를 잠재적 보안 위협이 높은 태스크로 정하고 정보 수집 및 탐지를 수행한다. 제안 모듈은 공격 가능성이 있는 uid와 자격증명의 불일치가 최초로 발생한 부모 태스크 pid 및 현재 시스템호출을 요청한 pid를 수집하여 의심 리스트를 작성한다.

Fig. 4에 표현된 제안 모듈의 동작은 먼저 자격증명을 점검하여 일관성과 권한 상승 유무를 확인한다(1). 만약 일관성에 문제가 없다면 의심 리스트와 비교를 수행하여, 공격 이후에 발생할 수 있는 우회의 가능성을 배제한다. 이를 통해 공격 이후 자격증명을 새로 발급받는 경우를 탐지할 수 있다. 반대로 일관성과 권한 상승 문제가 있다면 최초 발생 여부를 확인하고 리스트에 추가한다(2). 이후에는 로그를 남기고, 보안 정책에 따른 처리를 수행한다(3). 정책은 문제가 발생할 수 있는 태스크에 종료 시그널을 전송하여 즉시 중단 또는 관리자에게 알람 전송, 마지막으로 둘 모두 수행하는 형태로 구성되어있다.

4.2 모듈의 탐지 동작 실험

모듈의 탐지가 정상적으로 수행되는지 여부를 확인하기 위해 다음과 같은 5가지 경우를 이용하여 실험을 수행하였다. 모든 실험에는 인증을 정상적으로 수행한 관리자 쉘과 일반 사용자 쉘을 동시에 실행하여, 공격에 의한 쉘의 동작과 정상 쉘을 다중 프로세스 환경에서 탐지 가능한지 여부를 확인하였다.

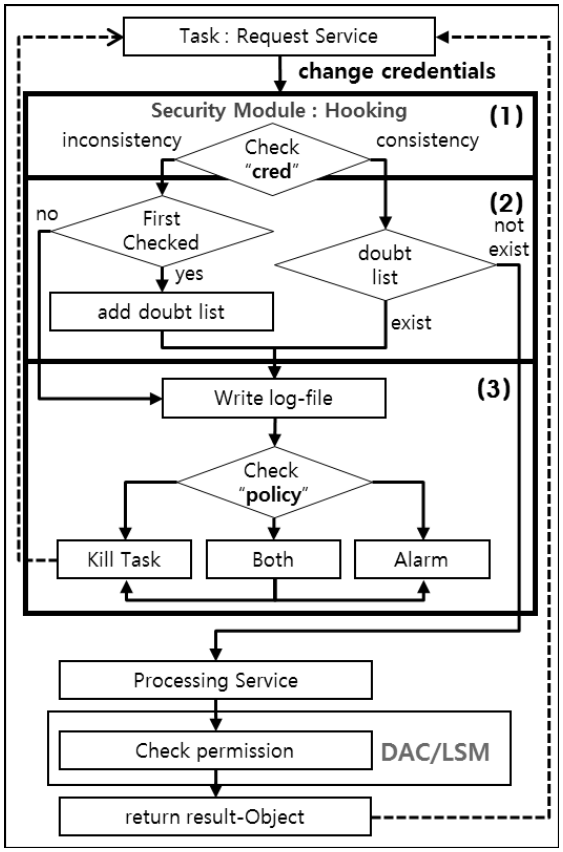


Fig. 4. The entire Module Process Flow

- case1 : user1 > root
- case2 : user1 >> root
- case3 : user1 > user2 >> root
- case4 : user1 >> root > root
- case5 : user1 >> root > root > root

탐지를 위해 실험을 수행한 환경은 다음과 같다.

- CPU : Intel Core i5 3.0GHz
- OS : Cent OS 6.3
- Kernel : 2.6.32-279

일반 사용자는 user1(uid/gid/groups:500), user2(505)로, 관리자는 root(0)으로 실험을 수행하였다. “su” 명령어를 이용하여 새로운 자격증명을 발급한 경우는 “>”으로, 공격을 통해 쉘을 실행한 경우는 “>>”으로 표현하였다. 탐지 결과는 아래 표와 같다.

표에 나타난 각각의 결과에서 2, 3번의 실험에서는 groups의 정보가 counter의 개수와 다름이 확인되었고, 이를 정상적으로 탐지하였다. 4, 5번 실험의 경우, 공격을 통해 관리자의 자격증명을 갖는 사용자가 “su” 명령어를 실행한 것으로, 정상적인 관리자 쉘과 자격증명이 발행되었다. 그러나 탐지를 위해 수집된 정보를 토대로 부모 태스크의 pid를 추적하

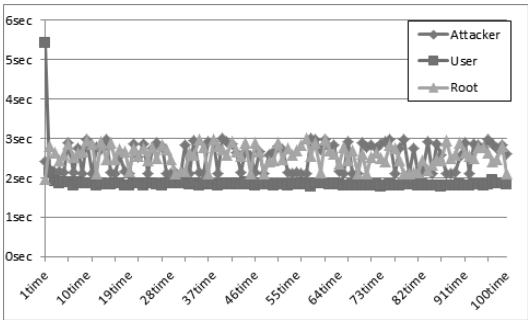
여, 자격증명이 불일치했던 시점의 부모 pid가 존재함을 확인할 수 있기 때문에 탐지가 가능하였다. 따라서 정상적인 관리자 자격증명을 갖는 쉘을 가지고 있더라도 태스크의 관계 정보를 이용하여 보안 위협의 발생이 가능한 태스크로 판단할 수 있었다. 또한 동시에 동작하는 정상 관리자의 쉘은 “su”를 통한 자격증명 변경 쉘의 동작과 혼동되지 않고 정상적으로 동작하였다. 결과적으로 자격증명의 불일치를 통해 작성된 의심 리스트를 이용하면 자격증명을 새로 할당받는 경우에도 탐지가 가능함을 확인하였다.

5. 성능 평가 및 우회 가능성

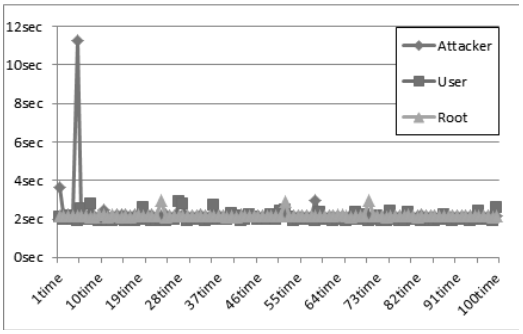
5.1 성능 평가

모듈 성능 평가는 전체 디렉터리에 존재하는 파일을 open(), close() 시스템호출을 이용하여 접근을 수행하고 이에 따른 경과 시간을 평가 기준으로 삼았다. 먼저 모듈을 적재하지 않은 상태에서 일반 사용자, 정상 관리자, 공격 관리자가 접근을 수행한 결과는 Fig. 5A와 같다. 다음으로 모듈을 적재한 상태에서 접근을 수행한 결과는 Fig. 5B와 같다. 전체 접근 시간을 모듈의 적재하지 않은 경우와 적재한 경우 각각 100회 반복 수행하여 집계하였다.

모듈을 적재하지 않은 상태에서 사용자는 평균 2.05초, 정상 관리자는 2.17초, 공격 관리자는 2.28초의 시간이 걸렸다. 모듈을 적재한 상태에서는 사용자가 평균 1.87초, 정상 관리자는 2.56초, 공격 관리자는 2.52초의 시간이 걸렸다. 사용자는 접근 권한이 없는 파일에 대한 접근이 거부되므로 다른 두 항목보다 더 적은 시간이 걸렸다. 모듈을 적재하지 않은 상태에서는 평균적으로 0.10초의 차이가 발생하였고, 적재한 상태에서는 0.04초의 차이가 발생하였다. 즉, 시스템호출 후킹을 통한 정보의 수집과 탐지는 시스템에 대한 부하가 미미함을 알 수 있다.



(A) Module unloaded



(B) Module loaded

Fig. 5. Performance of Experiment

5.2 모듈의 우회 가능성

본 논문이 제시한 groups의 정보 불일치를 이용한 탐지는 setgroups()를 통한 무력화 가능성이 존재한다. 그러나 탐지 실험에서 알 수 있듯이 자격증명이 변경된 이후, 정상적인 자격증명을 할당받는 경우에도 탐지가 가능하였다. 또한 setgroups()의 실행과정을 “strace”[16]로 추적한 결과, execve()을 시작으로 open() 2회, close() 2회 등의 파일처리 시스템호출 실행 후 실제 setgroups()가 동작하는 것을 확인할 수 있다[Fig. 6]. 즉, 공격자가 목표한 관리자 권한의 획득 후, setgroups()를 통해 모듈을 우회하려면, 제안된 모듈이 후킹하는 execve()와 파일처리 시스템호출을 이용해야 함을 알 수 있다. 따라서 setgroups()를 이용한 모듈의 우회는 불가능함을 알 수 있다.

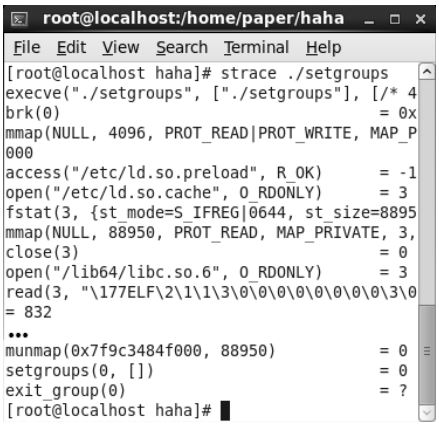


Fig. 6. Trace of setgroups() System Call Using “strace” Command

6. 결 론

본 논문이 제안한 탐지 모듈은 인증된 사용자의 자격증명과 태스크 실행이 발생시키는 특징을 통해 정보의 불일치가 발생하는 pid를 추적하여 자격증명이 불일치하는 태스크를 탐지할 수 있었다. 탐지를 위한 정보는 자격증명의 문제가 발생하는 uid와 실행 태스크의 pid값을 이용하였다. 이를 통

Table 1. Detecting Results

	case1	case2	case3	case4	case5
(e)uid	0	0	0	0	0
(e)gid	0	0	0	0	0
groups	0	500, 0	505, 0	0	0
counter	1	1	1	1	1
detect	N/A	O	O	O	O

해 작성된 의심 리스트는 이후의 모든 파일에 대한 접근을 수행하는 태스크의 pid와 비교하여 해당 태스크가 잠재적인 보안 위협을 발생할 수 있는지 여부를 판단하는 기준으로 사용하였다. 이를 통해 자격증명의 변경을 통한 우회와 시스템호출의 후킹을 이용한 우회를 막을 수 있었다. 또한 로그를 통한 수동적 보안 기초 정보와 공격자가 실행하는 태스크에 대한 정보 수집이 가능하다. 결과적으로 탐지 모듈은 시스템을 방해하지 않도록 부하를 최소화하면서, 커널의 수정 없이 정보를 수집하여 비정상적 자격증명을 탐지하였기 때문에 목표를 달성하였다고 볼 수 있다.

앞으로의 연구는 자격증명 일관성에 대한 처리를 수행하는 모듈의 특징을 이용하여 관리자뿐만 아니라 일반 사용자보다 높은 권한을 갖는 다른 권한들에 대해서도 탐지가 가능함을 타진하고, 더 많은 분야에 활용할 수 있도록 연구해야 할 것이다.

Reference

[1] Johri, Abhai, and Gary L. Luckenbaugh, “Trusted path mechanism for an operating system,” U.S. Patent No. 4,918,653, 17 Apr., 1990.

[2] SCARFONE, Karen; MELL, Peter. Guide to intrusion detection and prevention systems(idps). NIST special publication, 2007, 800.2007: 94.

[3] Ozdoganoglu, Hilmi, et al., “SmashGuard: A hardware solution to prevent security attacks on the function return address,” Computers, IEEE Transactions on 55.10(2006): 1271-1285.

[4] RICHARTE, Gerardo, et al. Four different tricks to bypass stackshield and stackguard protection. World Wide Web, http://www1.corest.com/files/files/11/StackGuardPaper.pdf, 2002.

[5] http://www.exploit-db.com/wp-content/themes/exploit/docs/27657.pdf

[6] Cowan, Crispin, et al., “StackGuard: Automatic adaptive detection and prevention of buffer-overflow attacks,” *Proceedings of the 7th USENIX Security Symposium*, Vol. 81, 1998.

[7] Ju-Hyuk Kim, Soo-Hyun Oh, “Detection Mechanism against Code Re-use Attack in Stack region,” *Journal of the Korea Academia-Industrial cooperation Society,* Vol.15 No.5, pp.3121-3131, 2014.

[8] TRAN, Minh, et al., On the expressiveness of return-into-libc attacks. In: Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, pp.121-141, 2011.

[9] LETOU, Kopelo; DEVI, Dhruwajita; SINGH, Y. Jayanta. Host-based Intrusion Detection and Prevention System (HIDPS), *International Journal of Computer Applications*, 69.26: 28-33, 2013.

[10] GOVINDARAJAN, M.; CHANDRASEKARAN, R. M. Intrusion detection using neural based hybrid classification methods, Computer networks, 55.8: 1662-1671, 2011.

[11] http://lwn.net/Articles/262464

[12] https://www.kernel.org/doc/Documentation/security/credentials.txt

[13] http://www.linfo.org/su.html

[14] http://en.wikipedia.org/wiki/System_call

[15] Wright, Chris, et al. “Linux security module framework.” Ottawa Linux Symposium. Vol.8032. 2002.

[16] McGrath, R. and W. Akkerman, “Source Forge Strace Project,” 2004.

김 원 일

e-mail : unangel@konkuk.ac.kr
2005년 건국대학교 컴퓨터공학과(학사)
2007년 건국대학교 컴퓨터공학과(석사)
2011년~현 재 유한대학 컴퓨터정보과 강의전담교원
관심분야 : 보안, 시스템 프로그래밍, 운영체제

유 상 현

e-mail : simonyoo@konkuk.ac.kr
2003년 Royal Holloway University of London Computer Science
2006년 건국대학교 컴퓨터공학과(석사)
2013년 건국대학교 컴퓨터공학과(박사)
현 재 (주)아이지코 연구원
관심분야 : 인공지능, 이미지 처리, 네트워크 보안

곽 주 현

e-mail : decoz91@gmail.com
1995년 건국대학교 전자계산학과(학사)
1997년 건국대학교 컴퓨터공학과(석사)
2012년 건국대학교 컴퓨터공학과(박사)
현 재 (주)아이지코 연구원
관심분야 : 인공지능, 소프트웨어 공학, 이미지 처리, 정보보안

이 창 훈

e-mail : chlee@konkuk.ac.kr
1977년 연세대학교 수학과(학사)
1980년 한국과학기술원 전산학과(석사)
1993년 학국과학기술원 전산학과(박사)
1996년~2002년 건국대학교 정보통신원장
2001년~2002년 건국대학교 정보통신대학과 학장
1980년~현 재 건국대학교 컴퓨터공학과 교수
관심분야 : 인공지능, 운영체제, 정보보안