KIPS Trans. Comp. and Comm. Sys.
Vol.9, No.7 pp.149~156 pISSN: 2287-5891

https://doi.org/10.3745/KTCCS.2020.9.7.149

A Study on Impersonation Attack of Linux Sudoers Through Shadow File Manipulation

Sanghun Kim† · Taenam Cho††

ABSTRACT

All operating systems have privileged administrator accounts for efficient management. Dangerous or sensitive tasks or resources should be banned from normal users and should only be accessible by administrators. One example of this privilege is to reset a user's password when the user loses his/her password. In this paper, the privileges of the sudoer group, the administrator group of Linux Ubuntu, and the management system of the sudoer group were analyzed. We show the danger that a sudoer can use the privilege to change the password of other users, including other sudoers, and modify the log, and suggest a countermeasure to prevent the manipulation of shadow files as a solution to this. In addition, the proposed method was implemented and the possibility of practical use was confirmed with excellent performance.

Keywords : Linux, Ubuntu, Password, Shadow File, Administrator Privilege

Shadow 파일 조작을 통한 리눅스 Sudoer의 위장공격에 대한 연구

김 상 훈† · 조 태 남††

요 약

모든 운영체제는 효율적인 관리를 위해 특권을 부여받은 관리자 계정이 존재한다. 위험하거나 민감한 작업이나 리소스는 일반 사용자에게는 접근이 허용되지 않아야 하며 오직 관리자에게만 허용되어야 한다. 이러한 특권의 한 가지 예는 사용자가 패스워드 분실하였을 때 초기화하는 권한이다. 본 논문에서는 리눅스 우분투의 관리자 그룹인 sudoer 그룹의 특권을 분석하고 관리자 그룹의 관리체계를 분석하였다. sudoer가 특권을 이용하여 다른 sudoer를 포함한 다른 사용자의 패스워드를 변경하고 로그를 수정함으로써 위장할 수 있는 위험성을 보이고, 이를 해결하기 위한 방안으로서 shadow 파일의 조작을 금지하는 방안을 제안하였다. 또한 제안한 방법을 구현하였으며 우수한 성능으로 실용화 가능성을 확인하였다.

키워드 : 리눅스, 우분투, 패스워드, 쉐도우 파일, 관리자 권한

1. 서 론

리눅스(Linux)는 전세계적으로 가장 많이 사용되는 무료 운영체제이다[1,2]. 리눅스 배포판은 다양하며 우분투(Ubuntu), 데비안(Debian), 센트OS(CentOS) 순으로 많이 사용되고 있다[3,4]. 리눅스를 포함하여 다중 사용자를 지원하는 모든 운영체제는 시스템 관리를 위한 관리자 계정이 존재한다. 이러한 관리자 계정은 시스템의 설치, 백업, 장치관리 및 일반 사용자 계정의 관리를 위하여 일반 계정에게 부여되지 않는 특권이 주어진다. 리눅스의 최고 관리자 계정은 root로서 누구나에게 공개되어 있으며, 그로 인해 모든 권한을 가진 시스템 관리자 계정은 공격자들의 대상이 되고 있다. 패스워드는 각 시스템마다 다르겠지만, 관리자 계정의 아이디와 패스워드 중 하나가 노출된 셈이기 때문에 우분투에서는 root 계정으로의 직접적 접속을 금지하여 공격자로부터 보호하고, root 대신 sudoer 그룹에 속한 사용자 계정으로 접속하여 sudo 명령을 통해 제한된 명령을 수행할 수 있도록 하고 있다.

관리자는 사용자의 계정 생성을 포함하여 거의 모든 작업을 수행할 수 있으며 사용자의 패스워드도 설정할 수 있다. 하지만, 시스템에 저장된 사용자들의 패스워드는 일방향 해시함수를 적용하여 저장되기 때문에 공격자로부터 보호될 뿐만 아니라 관리자도 다른 사용자의 패스워드를 알 수 없도록 하고 있다.

※ 이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2017R1D1A3B03032637).
† 비 회 원 : 우석대학교 정보보안학과 학사과정
†† 종신회원 : 우석대학교 IT전자융합공학과 교수
Manuscript Received : March 16, 2020
Accepted : April 24, 2020
* Corresponding Author : Taenam Cho(tncho@ws.ac.kr)

그러나 우리는 [5]에서 sudoer가 패스워드 저장방법의 특성을 이용하여 다른 사용자로 위장할 수 있음을 보이고, 커널을 수정하여 passwd 이외의 명령은 shadow 파일을 수정하지 못하도록 일부를 구현하여 대응책의 가능성을 보였다. 본 논문에서는 널리 사용되는 우분투에서 로그 삭제 등 sudoer가 수행할 수 있는 좀 더 정교하고 위험한 불법적인 행위를 보이고, [5]에서 제시한 대응책을 보완하고 구현하였다.

2. 우분투 시스템 관리 구조

2.1 sudoer 및 sudo

1) 계정과 그룹

리눅스 시스템에서는 모든 권한을 가지는 root라는 관리자 계정이 디폴트로 존재한다. 우분투에서는 시스템 설치 시 root 외에 관리자 계정을 생성하고 패스워드를 설정한다. 이후의 계정은 이 관리자 계정을 통해 생성하게 된다. 계정을 생성하면 동시에 계정명과 동일한 그룹이 생성되는데, 모든 일반 사용자는 계정과 동일한 이름의 1개 그룹에 속하게 된다. 단, 설치 시 생성된 관리자 계정은 디폴트로 여러 개의 계정에 속하게 된다. 시스템에 생성된 그룹은 /etc/group 파일에 정의되어 있다. 예로, 설치 시 생성된 관리자 계정이 topAdmin일 경우, root 그룹의 gid는 0, sudo 그룹의 gid는 27, topAdmin 그룹의 gid는 1000이 할당된다. 그 외에도 많은 디폴트 그룹들이 존재하는데, topAdmin은 sudo, adm, cdrom, plugdev, sambashare, lpadmin과 같은 주요 그룹에 소속된다. 우리는 모든 권한을 가지는 sudo 그룹에 초점을 맞출 것이다.

2) sudoer

우분투에도 root 계정이 존재하기는 하지만 보안을 위해 root 계정으로 로그인을 허용하지 않는다. 대신 root와 유사한 권한을 갖는 sudo 그룹을 유지한다. 이 그룹에 속한 사용자들을 sudoer라고 하며 기존의 root 역할을 하게 된다. /etc/sudoers라는 파일에서 그룹이나 계정별로 허용되는 명령을 제한할 수 있다. Fig. 1은 sudo 그룹에 속한 사용자는 모든 명령을 실행할 수 있도록 설정된 /etc/sudoers 파일의 일부이다. 이 파일은 sudoer가 sudo 명령을 통해 visudo라는 명령을 이용해야만 수정이 가능하도록 함으로써 보호하고 있다.

```
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

Fig. 1. Privilege of Sudo Group

3) sudo

계정의 생성/삭제와 패스워드 설정과 같은 주요 명령(privileged command)은 “sudo”라는 명령을 통해서만 실행할 수 있다. sudo와 함께 명령을 실행시키면 Fig. 2와 같이 실행하고 있는 계정의 패스워드를 다시 한 번 요구한다. 패스워드가 일치하면 실행 계정이 sudo 그룹에 속해 있는지 확인하여 실행을 허용한다[6].

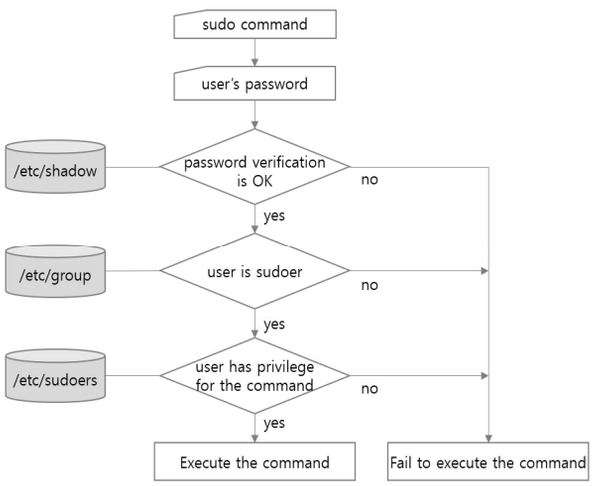

Fig. 2. Process for Sudo Command

Fig. 3은 sudoer인 topAdmin이 sudo 명령을 통하여 sudoer1 계정을 생성하는 과정이다.

```
topadmin@ubuntuo:~/Desktop$ sudo adduser sudoer1
[sudo] password for topadmin:
Adding user `sudoer1' ...
Adding new group `sudoer1' (1001) ...
Adding new user `sudoer1' (1001) with group `sudoer1' ...
Creating home directory `/home/sudoer1' ...
Copying files from `/etc/skel' ...
New password:
```

Fig. 3. User Account Creation via sudo Command

만약, Fig. 4와 같이 sudoer가 아닌 user1이 sudo를 사용할 경우에는 오류 메시지와 함께 실행이 중지된다.

```
user1@ubuntu0:~/Desktop$ sudo adduser user2
[sudo] password for user1:
user1 is not in the sudoers file.  This incident will be reported.
user1@ubuntu0:~/Desktop$
```

Fig. 4. Deny sudo Command from End User

2.2 패스워드 관리

패스워드는 계정의 보호와 시스템 접근권한을 제어하기 위한 가장 기본적인 인증 수단이다. 각 사용자의 패스워드는 /etc/shadow라는 파일에 저장되며[7], "passwd"라는 명령을 통해서 패스워드를 변경함에 따라 수정된다[8]. 일반 사용자는 자신의 패스워드만 변경이 가능하며, sudoer는 다른 사용자의 패스워드도 변경할 수 있다. shadow 파일에는 각 계

정마다 여러 가지 필드들이 콜론(:)으로 구분되어 저장된다. 첫 번째 필드는 계정명이고 두 번째 필드는 패스워드이다. 패스워드는 평문으로 저장하지 않고 역계산이 불가능한 일방향 해시함수를 적용하여 저장한다. 또한 사전공격을 막기 위해서, 동일한 패스워드라도 다른 값으로 저장되도록 시스템이 생성한 랜덤수인 salt와 함께 해시한다. 패스워드 필드는 Table 1과 같이 $로 구분되는 3개의 서브필드로 구성된다. 지정한 해시함수가 H이라면 hashed_password = H(user_password || salt)와 같이 계산된다.

Table 1. Password Field of Shadow File

Sub Field	Meaning
Hash_id	Applied hash function's id
Salt	random number generated by system
Hashed_password	hash value of user's password and salt

Fig. 5는 topAdmin의 저장된 패스워드이다. hash_id 6은 SHA_256을 해시함수로 사용한다는 의미이다.

```
topadmin:$6$fVWQZayqwc4MsEEN$Q3KEwHrpfMncPyT74wRkvJWA2U4G6f5.Fr
GaeyTqTh6MQJvectObXWQ4nMslmKqQSw6PkTvOg2UYWeCEr/EOt/:18333:0:99
999:7:::
```

Fig. 5. Shadow File for topAdmin

사용자가 로그인할 때 아이디와 패스워드를 입력하면, Fig. 6에서와 같이 /etc/shadow 파일로부터 적용할 해시함수를 결정하고 salt를 입력된 패스워드에 접합하여 해시한다. 그 결과가 /etc/shadow 파일에 저장된 값과 일치하면 인증이 성공한 것으로 간주하여 로그인 처리를 한다.

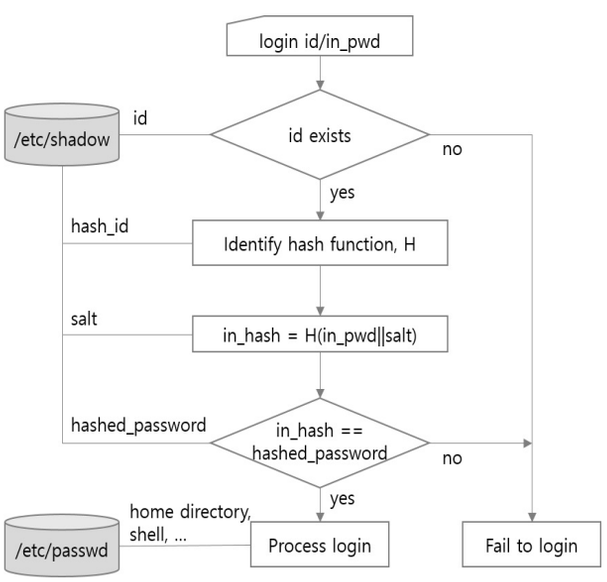

Fig. 6. Authentication using Login Password

2.3 로그 관리

우분투에는 시스템에서 수행되는 동작에 대해 다양한 로그가 저장된다. 이벤트의 발생 주체와 위험도에 따라 수집할 로그를 설정할 수 있지만, 디폴트로 sudo 명령 사용에 대해서는 /var/log/auth.log에 기록된다. auth.log도 일반 사용자는 읽기만 가능하며 sudoer만이 수정할 수 있다. 기타의 일반 명령에 대한 사용은 각 계정의 홈디렉토리에 있는 .bash_history라는 파일에 남게 된다.

3. 공격 시나리오

3.1 취약점 분석

1) sudoer 권한

sudoer는 모든 명령을 사용할 수 있으며 모든 파일에 접근이 가능하다. 따라서 sudoer는 passwd 명령을 사용하지 않고도 shadow 파일을 직접적으로 읽거나 수정할 수 있으며 log 파일도 수정하여 auth.log에서 sudo 로그를 삭제할 수 있다. 다른 sudoer들과 달리 root는 sudo를 사용하지 않고도 특권 명령을 사용할 수 있기 때문에 auth.log에 기록조차 남지 않는다.

2) shadow 파일

shadow 파일에 저장되는 패스워드는 사용자가 설정한 패스워드와 시스템이 생성한 난수 salt를 접합하여 해시함수에 적용한 결과이다. 해시함수는 해시값으로부터 원래의 입력값을 알 수 없다는 일방향성으로 인하여 보안에서 많이 사용된다. sudoer들은 다른 사용자의 패스워드를 알 수는 없지만 새로 설정할 수 있다. 하지만 sudoer가 다른 사용자의 패스워드를 임의로 새로 설정할 경우, 사용자가 이를 인지할 수 있기 때문에 함부로 바꾸지 못한다.

문제는 동일한 해시함수에 대하여 입력값이 동일할 경우 결과값도 동일하며, sudoer들은 shadow 파일을 직접 수정할 수 있다는 데에 있다. salt 값은 passwd를 통하여 패스워드를 변경할 때는 새로운 값으로 갱신되지만, 파일을 직접 수정하거나 쓰기 및 덮어쓰기 할 때는 salt 값이 갱신되지 않는다. 따라서 shadow 파일을 수정할 권한을 가진 sudoer가 shadow 파일에서 사용자의 패스워드 필드를 복사해 두었다가 새 패스워드로 변경한 후, 다시 이전 패스워드 필드로 복구하면 이전 상태로 돌아가기 때문에 사용자는 이를 알아채지 못한다.

3.2 sudoer의 위장 공격

sudoer의 공격시나리오는 Fig. 7과 같다. ① sudoer는 주어진 특권을 이용하여 shadow 파일을 복사하거나, shadow 파일에서 희생자의 패스워드 필드(hash_id, salt와 hashed_password)을 복사해 놓은 후, ② 희생자의 패스워드를 새로

설정한다. ③ 그런 다음 희생자의 계정으로 로그인하여 부적절한 행위를 수행한 후, ④ 다시 sudoer로 로그인하여 원래의 shadow 파일로 복구하거나 shadow 파일을 수정하여 원래의 패스워드 필드로 복구해 놓는다. 희생자는 자신이 설정한 패스워드로 설정되어 있기 때문에 패스워드의 변경사실을 인식하지 못한다[5]. ⑤ sudoer는 sudo 명령 사용에 대한 시스템 로그를 삭제한다.

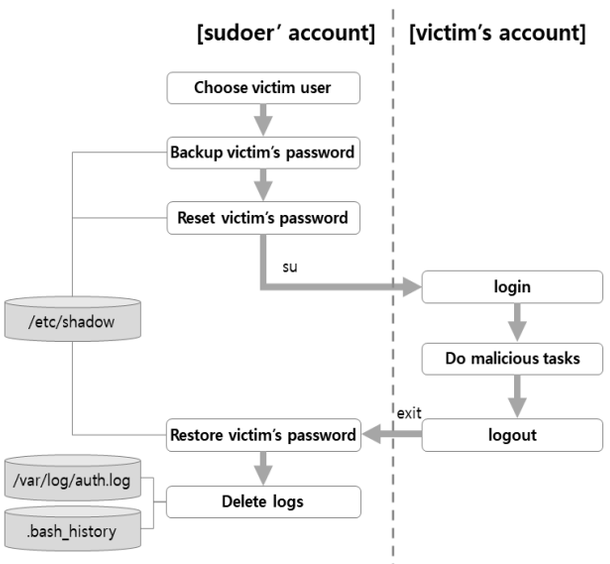

Fig. 7. Impersonate Scenario of Sudoers

Fig. 8은 topAdmin이 shadow 파일을 백업한 후 공격 대상자인 user1의 패스워드를 변경했다가 restore한 후, user1이 이전의 패스워드로 로그인에 성공하는 것을 보여준다. 물론 예에서는 테스트를 위하여 topAdmin이 user1의 이전 패스워드를 알고 있어 "su user1"으로 로그인하였다.

```
topadmin@ubuntuO:~/Desktop$ sudo cp /etc/shadow orgShadow
[sudo] password for topadmin:
topadmin@ubuntuO:~/Desktop$ sudo passwd user1
New password:
Retype new password:
passwd: password updated successfully
topadmin@ubuntuO:~/Desktop$ su user1
Password:
user1@ubuntuO:/home/topadmin/Desktop$ exit
exit
topadmin@ubuntuO:~/Desktop$ sudo mv orgShadow /etc/shadow
topadmin@ubuntuO:~/Desktop$ su user1
Password:
user1@ubuntuO:/home/topadmin/Desktop$
```

Fig. 8. Successful Login of Victim After Restore Shadow File

shadow 파일의 수정이나 복사 등은 sudo 명령을 통해서만 가능하기 때문에 sudoer의 공격 과정은 Fig. 9와 같이 auth.log에 로깅되었다. 그러나 이 로그 또한 sudoer는 직접 수정이 가능하다. Fig. 10은 해당 로그를 삭제한 후의 auth.log를 보여준다.

```
topadmin@ubuntuo:/etc$ sudo cp shadow backup_shadow
topadmin@ubuntuo:/etc$ sudo cp backup_shadow shadow
topadmin@ubuntuo:/etc$ sudo cat /var/log/auth.log | tail -10
May  5 02:57:40 UbuntuM sudo: pam_unix(sudo:session): session
opened for user root by (uid=0)
May  5 02:57:40 UbuntuM sudo: pam_unix(sudo:session): session
closed for user root
May  5 02:58:33 UbuntuM sudo: topadmin : TTY=pts/0 ; PWD=/etc
; USER=root ; COMMAND=/usr/bin/cp shadow backup_shadow
May  5 02:58:33 UbuntuM sudo: pam_unix(sudo:session): session
opened for user root by (uid=0)
May  5 02:58:33 UbuntuM sudo: pam_unix(sudo:session): session
closed for user root
May  5 02:58:37 UbuntuM sudo: topadmin : TTY=pts/0 ; PWD=/etc
; USER=root ; COMMAND=/usr/bin/cp backup_shadow shadow
May  5 02:58:37 UbuntuM sudo: pam_unix(sudo:session): session
opened for user root by (uid=0)
May  5 02:58:37 UbuntuM sudo: pam_unix(sudo:session): session
closed for user root
May  5 02:58:44 UbuntuM sudo: topadmin : TTY=pts/0 ; PWD=/etc
; USER=root ; COMMAND=/usr/bin/cat /var/log/auth.log
May  5 02:58:44 UbuntuM sudo: pam_unix(sudo:session): session
opened for user root by (uid=0)
topadmin@ubuntuo:/etc$
```

Fig. 9. Sudo Command Logs Stored in Auth.log

```
topadmin@ubuntuo:/etc$ sudo vi /var/log/auth.log
topadmin@ubuntuo:/etc$
topadmin@ubuntuo:/etc$ cat /var/log/auth.log | tail -5
May  5 02:58:37 UbuntuM sudo: pam_unix(sudo:session): session
closed for user root
May  5 02:58:44 UbuntuM sudo: pam_unix(sudo:session): session
opened for user root by (uid=0)
May  5 02:58:44 UbuntuM sudo: pam_unix(sudo:session): session
closed for user root
May  5 02:59:35 UbuntuM sudo: pam_unix(sudo:session): session
opened for user root by (uid=0)
May  5 03:00:10 UbuntuM sudo: pam_unix(sudo:session): session
closed for user root
topadmin@ubuntuo:/etc$
```

Fig. 10. Auth.log After Delete Sudo Logs

만약 일반 사용자가 아니라 다른 sudoer를 희생자로 선택한다면, 좀 더 심각한 문제를 야기할 수 있다. 일반 사용자보다 sudoer는 훨씬 많은 일을 할 수 있고, 공격자가 원하는 불법적 sudo 작업을 다른 sudoer의 계정으로 수행할 수 있으며, 수행한 sudo 작업들이 고스란히 auth.log에 로그가 남는다.

또 다른 위험성은 악의적인 sudoer가 root 계정으로도 작업할 수 있다는 것이다. sudoer는 root의 역할을 할 수 있다는 점에서는 동일하게 취급할 수 있으나, root 계정으로 작업할 때는 sudo를 이용하지 않아도 되기 때문에 auth.log에도 로그가 남지 않는다는 점이 다르다. 따라서 악의적인 sudoer가 root로 위장한 다음 로그아웃하기 전에 sudoer로 작업했던 패스워드 조작 로그를 삭제하고 syslog 서비스를 재시작하면 서비스 재시작 기록만 남게 된다.

따라서 Fig. 7에서와 같이 희생자 계정에서 로그아웃하고 다시 sudoer로 로그인했을 때, auth.log에 기록된 shadow 파일 조작 로그와 auth.log를 수정하는 로그를 삭제하면 Fig. 10과 같이 어떤 작업을 했는지는 알아낼 수 없다. 단, 로그를 수정할 경우 syslog 서비스를 재시작 해야 하기 때문에 그 기록만이 남을 뿐이다.

4. 대응 방안

지금까지 살펴본 공격이 가능한 원인은 2가지이다. 첫째, 입력값이 동일하면 해시함수는 항상 동일한 해시값을 산출한다는 것이다. 둘째, sudoer가 passwd 명령을 통하여 다른 사용자의 패스워드를 새로운 값으로 설정하는 것으로 그치지 않고 shadow 파일을 복사, 삭제, 덮어쓰기, 그리고 에디터를 이용한 수정이 가능하기 때문이다. 따라서 두 가지 조건 중의 하나를 제거하면 해결될 수 있다. 동일한 입력에 대해 예측할 수 없는 해시값을 생성하는 해시함수를 사용할 수는 없다. 따라서 첫 번째 조건을 없애기 위해서는 패스워드가 변경될 때마다 다른 salt가 생성되도록 하는 것이나 매우 복잡한 구조가 필요하다. 따라서 본 논문에서는 두 번째 조건인 패스워드의 의심스러운 조작을 방지하도록 한다.

4.1 요구사항

1) 직접적 접근 제한: shadow 파일에 대하여 root를 포함하여 sudoer의 다음과 같은 직접적인 접근을 제한해야 한다.
   - root나 sudoer도 에디터를 이용하여 shadow 파일을 수정할 수 없다.
   - root나 sudoer도 shadow 파일을 삭제, 덮어쓰기, 이름변경을 할 수 없다.

2) 정상 접근 허용: shadow 파일을 사용하는 합법적 접근을 허용해야 한다.

4.2 핵심 대응 아이디어

요구사항을 만족하도록 커널을 수정하기 위하여 우선 shadow 파일 수정이 허용된 명령들를 분류하고, 우회적 수정할 가능성이 있는 명령을 포함하여 금지시켜야 할 명령들을 분류하여야 한다. 또한 명령을 통한 우회도 가능하지만 상대경로를 이용한 우회도 가능하므로, 이를 방지하기 위하여 다음과 같이 커널 수정 기준을 마련하였다.

- Table 2에 나열된 명령은 /etc/shadow 파일에 대한 수정을 허용한다.
- Table 3에 나열된 명령은 /etc/shadow 파일에 대한 수정을 금지한다.
- 상대경로를 이용하여 우회적으로 /etc/shadow 파일을 수정하지하지 못하도록 파일의 절대경로를 인식할 수 있어야 한다.
- 기타 /etc/shadow에 대한 허용된 읽기나 쓰기 명령은 기존의 접근권한을 준용한다.
- 성능 저하를 방지하기 위하여 커널의 수정을 최소화 한다.

Table 2. Permitted Commands

Command	Function
passwd	Change user password
login	Begin session on the system
cp	Copy files and directories
chage	Change user password expiry information
pwck	Verify integrity of password files
su	Run a command with substitute user and group ID
sulogin	Single-user login
adduser deluser	Create and delete user account
sudo	Run programs with the security privileges of the root

Table 3. Prohibited Commands

Command	Function
mv	Move(copy or write) and rename files
rm	Remove files and directories
touch	Create, change and modify timestamps of files
〉	Redirect output stream to a file
pwconv, pwunconv	Convert to and from shadow passwords
ed, ex, sed, vi, emacs, nano, gedit (Editors)	Edit files

pwconv와 pwunconv명령은 비정상적인 접근은 아니지만, 누구나 접근할 수 있는 passwd 파일로부터 shadow 파일을 생성하기 때문에 금지 명령으로 처리하였다. copy 명령은 shadow 파일에 대한 읽기만 요구하기 때문에 본 연구에서는 허용하도록 하였으나, 잠재적 위험요소를 없애기 위해서 금지하도록 할 수도 있다.

4.3 구현 및 실험

1) 구현

수정 구현은 최신버전인 Ubuntu 19.10을 사용하였다. 수정을 최소화하기 위하여 커널을 분석한 결과, Table 2에 나열한 허용된 명령들은 공통적으로 do_sys_open() 시스템 함수를 호출한다. 그러나 Table 3에 나열한 금지 명령들은 Table 4와 같이 다양한 시스템 호출을 하고 있기 때문에 이들 함수에 대한 수정이 요구된다. do_sys_open()에서는 허용되지 않은 명령이 etc/shadow을 수정하려고 할 때 이를 거부하도록 한다. real_path()에서는 상대경로를 통하여 우회하는 것을 금

Table 4. Prohibited Commands and Related System Calls

Command	Related system call
cp	stat(), do_sys_open()
mv	lstat(), stat(), rename()
rm	unlink()
touch	do_sys_open()
〉	do_sys_open()
pwconv, pwunconv	do_sys_open(), unlink()
ed, ex, sed, vi, emacs, nano, gedit (Editors)	do_sys_open(), stat(), unlink(), rename()

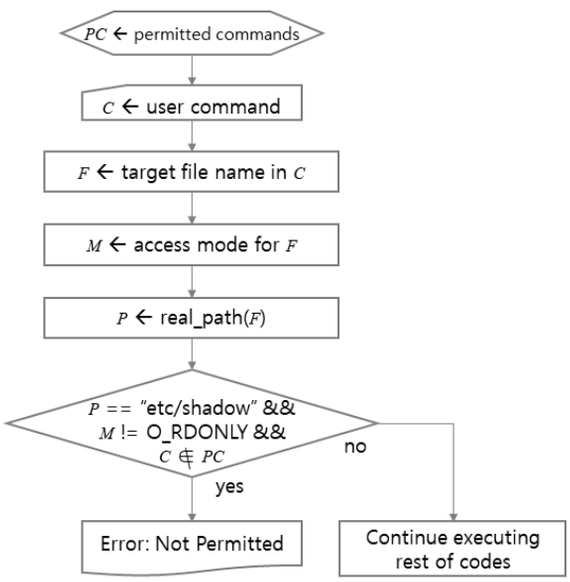

Fig. 11. Flowchart of Function Do_sys_open()

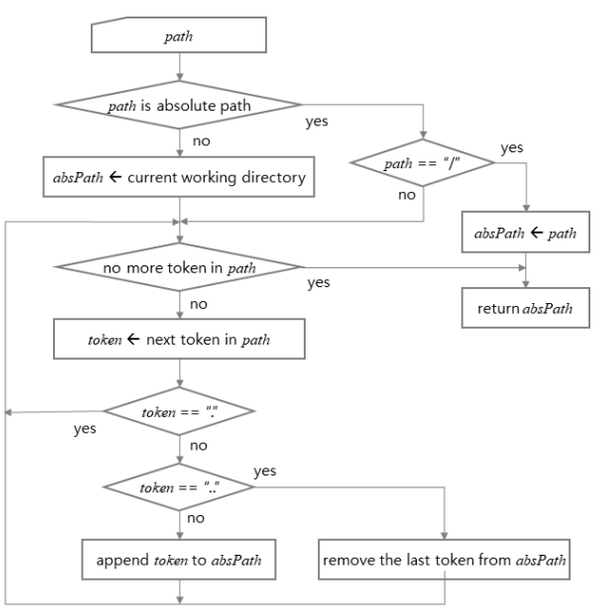

Fig. 12. Flowchart of Function Real_path()

```
topadmin@ubuntum:~$ # ----- remove with absolute path
topadmin@ubuntum:~$ sudo rm /etc/shadow
[sudo] password for topadmin:
rm: cannot remove '/etc/shadow': Permission denied
topadmin@ubuntum:~$
topadmin@ubuntum:~$ # ----- overwrite to /etc/shadow
topadmin@ubuntum:~$ touch newShadow
topadmin@ubuntum:~$ sudo mv newShadow /etc/shadow
mv: cannot move 'newShadow' to '/etc/shadow': Permission denied
topadmin@ubuntum:~$
topadmin@ubuntum:~$ # ----- rename
topadmin@ubuntum:~$ sudo mv /etc/shadow /etc/Shadow
mv: cannot move '/etc/shadow' to '/etc/Shadow': Permission denied
topadmin@ubuntum:~$
topadmin@ubuntum:~$ # ----- remove with relative path name
topadmin@ubuntum:~$ pwd
/home/topadmin
topadmin@ubuntum:~$ sudo rm ../../etc/shadow
rm: cannot remove '../../etc/shadow': Permission denied
topadmin@ubuntum:~$
topadmin@ubuntum:~$ # ----- edit and save with editor
topadmin@ubuntum:~$ sudo vi /etc/shadow
                    (중략)
avahi:*:18186:0:99999:7:::
saned:*:18186:0:99999:7:::
nm-openvpn:*:18186:0:99999:7:::
:wq
                    (중략)
saned:*:18186:0:99999:7:::
nm-openvpn:*:18186:0:99999:7:::
"/etc/shadow"
"/etc/shadow" E212: Can't open file for writing
Press ENTER or type command to continue
```

Fig. 13. Prohibited Command Execution in Modified System

지하기 위해 파일의 경로를 절대경로로 변환한다. 이 두 함수에 대한 처리흐름을 Fig. 11과 Fig. 12에 나타내었다. 나머지 함수들은 do_sys_open()과 유사하므로 기술을 생략한다.

2) 실험 결과

본 연구에서 수정한 우분투 시스템 UbuntuM에서 sudoer 계정인 topAdmin이 금지된 명령을 수행한 결과 Fig. 13에서와 같이 모두 오류로 인식하고 실행이 중지된다. 상대경로로 접근한 경우에도 접근이 제한되고 있다.

허용된 명령들에 대해서는 대표적인 명령의 수행 결과를 Fig. 14에 나타내었으며 정상적으로 수행되는 것을 볼 수 있다.

본 연구에서 구현한 시스템과 기존의 시스템의 성능을 비교하기 위하여 “time” 명령으로 허용된 명령들의 수행시간을 비교하였다. 이 명령은 실제 수행시간(real), 사용자 영역에서의 수행시간(user)과 시스템 영역에서의 수행시간(sys)을 초단위로 보여준다. 그러나 passwd와 같이 패스워드의 입력과 같은 사용자의 개입이 요구되는 경우에 실제 수행시간은 사용자의 입력시간까지 포함하기 때문에, 시스템 성능 비교에는 적합하지 않다. 따라서 Table 5에서 사용자 영역과 시스템 영역에서의 수행시간만을 비교하였다. 수행시간은 거의 차이가 없으며, 심지어 수정된 시스템이 더 좋은 성능을 나타내기도 하였다. 이러한 차이는 시스템 수행 당시의 다른 요소에 의한 미세한 차이로 판단된다.

```
topadmin@ubuntum:~$ # ----- browse (read)
topadmin@ubuntum:~$ sudo cat /etc/shadow
root:!:18333:0:99999:7:::
daemon:*:18186:0:99999:7:::
bin:*:18186:0:99999:7:::
                    (중략)
topadmin@ubuntum:~$ # ----- copy
topadmin@ubuntum:~$ sudo cp /etc/shadow backupShadow
topadmin@ubuntum:~$
topadmin@ubuntum:~$ ls backupShadow
backupShadow
topadmin@ubuntum:~$
topadmin@ubuntum:~$ # ----- add new user account user1
topadmin@ubuntum:~$ sudo adduser user1
Adding user `user1' ...
Adding new group `user1' (1003) ...
Adding new user `user1' (1003) with group `user1' ...
Creating home directory `/home/user1' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for user1
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
topadmin@ubuntum:~$
topadmin@ubuntum:~$ # ----- login as user1
topadmin@ubuntum:~$ su user1
Password:
user1@ubuntum:/home/topadmin$ exit
exit
topadmin@ubuntum:~$
topadmin@ubuntum:~$ # ----- delete use account user1
topadmin@ubuntum:~$ sudo deluser user1
Removing user `user1' ...
Warning: group `user1' has no more members.
Done.
topadmin@ubuntum:~$
```

Fig. 14. Permitted Command Execution in Modified System

Table 5. Comparison of Permitted Command Execution Time

Command	Original System		Modified System	
	User	System	User	System
passwd	0.040	0.006	0.039	0.004
cp	0.004	0.012	0.009	0.011
chage	0.005	0.018	0.006	0.016
pwck	0.009	0.013	0.011	0.010
su	0.048	0.019	0.057	0.016
sulogin	0.014	0.029	0.020	0.029
adduser	0.147	0.082	0.078	0.090
deluser	0.153	0.053	0.116	0.054

5. 결론 및 향후 연구

본 논문에서는 전 세계적으로 널리 사용되고 있는 리눅스 우분투의 관리자 관리 방법과 권한, 로그 관리 방식 및 패스워드 관리 방식을 분석하였다. 이를 토대로 우분투의 관리자 그룹인 sudoer가 주어진 권한을 이용하여 다른 일반 사용자 뿐만 아니라 root의 패스워드를 변경함으로써 발생할 수 있는 위장공격의 위험성을 보였다. 또한 sudoer의 불법적 행위가 추적될 수 있는 관련 로그도 조작이 가능한 것을 확인하였다. 이러한 sudoer의 위장공격에 대응하기 위하여 합법적 접근으로 허용해야 하는 명령들을 분류하여 이를 제외한 명령들은 shadow 파일에 접근하지 못하도록 커널을 수정하는 방법을 제안하였으며, 이를 구현하였다. 수정한 시스템은 수정으로 인한 성능저하를 보이지 않기 때문에 실용화가 가능할 것으로 판단된다. 본 실험은 CentOS에서도 동일하게 적용됨을 확인하였으며 동일 커널을 사용하는 다른 리눅스 배포판에서도 동일하게 적용될 수 있을 것으로 판단된다.

향후 추가되는 패스워드와 관련된 명령들에 대해서는 추가적인 허용여부를 판단하여 허용해야하는 명령의 경우는 허용 명령 집합에 추가하도록 커널을 수정해야 한다. 또한 shadow 파일과 마찬가지로 auth.log에 대한 의심스러운 수정을 금지하는 것도 보안을 강화하는 방법이 될 것이다.

References

[1] Linux [Internet], https://www.linux.org/

[2] Top 7 PCs Shared by World / Domestic [Internet], http://catalk. kr/information/desktop-operating-systems.html

[3] Ubuntu [Internet], https://ubuntu.com/

[4] Historical trends in the usage statistics of Linux subcategories for websites [Internet], https://w3techs.com/technologies/history_details/os-linux.

[5] S. Kim and T. Cho, "A Study on Vulnerabilities of Linux Password and Countermeasures," CUTE 2019. paper No.9, 2019.

[6] Michael Kerrisk, Linux Programmer's Manual [Internet], http://man7.org/linux/man-pages/man5/group.5.html. GitHub.

[7] Michael Kerrisk, File Formats and Conversions-SHADOW [Internet], http://man7.org/linux/man-pages/man5/shadow.5.html, GitHub.

[8] Michael Kerrisk, User Command-PASSWD [Internet], http://man7.org/linux/man-pages/man1/passwd.1.html, GitHub.

김 상 훈

https://orcid.org/0000-0002-6019-616X

e-mail : schmid_t@naver.com

2016년 ~ 현 재 우석대학교 정보보안학과 학사과정

관심분야 : Cryptography, System Security, System Programming

조 태 남

https://orcid.org/0000-0002-5191-0130
e-mail : tncho@ws.ac.kr
1986년 이화여자대학교 전자계산학과(학사)
1988년 이화여자대학교 전자계산학과(석사)
2004년 이화여자대학교 컴퓨터학과(박사)
1988년 ~ 1997년 한국전자통신연구원
위성관제연구실 선임연구원
2004년 ~ 2005년 이화여자대학교 컴퓨터학과 전임강사
2005년 ~ 2017년 우석대학교 정보보안학과 교수
2018년 ~ 현 재 우석대학교 IT전자융합공학과 교수
관심분야 : Android Security, Bluetooth, Block-Chain