資訊系統與作業安全

Linux 系統中的日誌記錄與稽核

羅勻瑄、黃世君、劉定睿、曾彥輔

日期：2026-04-28

目錄

第一題

- 作業目標:
  - 學習並理解 Linux 系統中的日誌記錄功能，透過觀察 /var/log/auth.log，熟悉系統如何記錄與身份驗證、權限操作相關的事件
- 具體目標如下：
  - 查找登入成功的紀錄
  - 查找登入失敗的紀錄
  - 查找使用 sudo 指令的相關紀錄

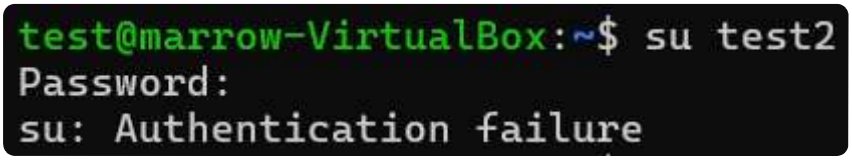
參考作法

1. 透過以下指令分別創造兩個測試帳號:test, test2

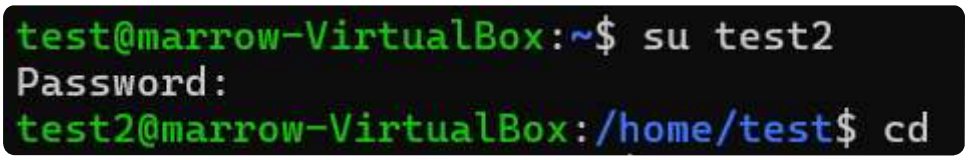
```
# 建立帳號並且給予家目錄和預設使用bash
sudo useradd -m -s /bin/bash test
sudo useradd -m -s /bin/bash test2

# 設定用於登入的密碼
sudo passwd test
sudo passwd test2
```

2. 製造登入成功與失敗的事件

登入test後，嘗試使用錯誤的密碼登入test2



再次嘗試登入test2，改為使用正確密碼並成功登入



3. 執行數個 sudo 指令來觸發權限操作紀錄

使用指令 `sudo -l` 來查看自己可用的指令

```
test2@marrow-VirtualBox:~$ sudo -l
[sudo] password for test2:
Sorry, user test2 may not run sudo on marrow-VirtualBox.
```

> ⚠ Warning
>
> 發現根本沒在sudoers名單上，因此無法使用sudo

使用管理權限帳號使用指令 `sudo visudo` 修改sudoers後，繼續嘗試使用sudo指令

1. 輸入 `sudo find . -exec /bin/sh \; -quit` 嘗試做壞壞的事情

   ```
   test2@marrow-VirtualBox:~$ sudo find . -exec /bin/sh \; -quit
   [sudo] password for test2:
   Sorry, user test2 is not allowed to execute '/usr/bin/find . -exec /bin/sh ; -quit' as root on marrow-VirtualBox.
   ```

   結果發現管理員沒有給自己執行的權限

2. 輸入 `sudo -l` 查看自己可以執行的指令

   ```
   test2@marrow-VirtualBox:~$ sudo -l
   Matching Defaults entries for test2 on marrow-VirtualBox:
       env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

   User test2 may run the following commands on marrow-VirtualBox:
       (ALL) NOPASSWD: /usr/bin/apt update, /usr/bin/apt upgrade
   ```

   結果發現管理員只讓自己更新東西而已...

3. 輸入 `sudo apt update` 乖乖幫忙更新

   ```
   test2@marrow-VirtualBox:~$ sudo apt update
   Hit:1 http://tw.archive.ubuntu.com/ubuntu noble InRelease
   Get:2 http://tw.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
   Hit:3 http://tw.archive.ubuntu.com/ubuntu noble-backports InRelease
   Get:4 https://dl.google.com/linux/chrome-stable/deb stable InRelease [1,825 B]
   Hit:5 https://packages.microsoft.com/repos/code stable InRelease
   Hit:6 http://security.ubuntu.com/ubuntu noble-security InRelease
   Get:7 https://dl.google.com/linux/chrome-stable/deb stable/main amd64 Packages [1,218 B]
   Get:8 http://tw.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,919 kB]
   Get:9 http://tw.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [345 kB]
   Get:10 http://tw.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2,981 kB]
   Hit:11 https://us-central1-apt.pkg.dev/projects/antigravity-auto-updater-dev antigravity-debian InRelease
   Get:12 http://tw.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [697 kB]
   Fetched 6,071 kB in 2s (3,426 kB/s)
   Reading package lists... Done
   Building dependency tree... Done
   Reading state information... Done
   43 packages can be upgraded. Run 'apt list --upgradable' to see them.
   ```

透過 cat 和 grep 指令來查看 auth.log

1. 輸入 `sudo cat /var/log/auth.log | grep -i "fail"`

   ```
   2026-04-22T17:12:22.035716+08:00 marrow-VirtualBox gdm-
   password]: pam_unix(gdm-password:auth): authentication
   failure; logname= uid=0 euid=0 tty=/dev/tty1 ruser= rho
   st=  user=test2
   ```

- 時間:2026-04-22T17:12:22
- 主機:marrow-VirtualBox
- 程式:gdm-password(ubuntu的圖形登入驗證介面)
- 事件:authentication failure
- 嘗試登入帳號:test2

2. 輸入 `grep "session opened" /var/log/auth.log | grep -v "sudo"`

```
2026-04-22T17:13:40.837803+08:00 marrow-VirtualBox gdm-
password]: pam_unix(gdm-password:session): session open
ed for user test2(uid=1003) by test2(uid=0)
```

- 時間:2026-04-22T17:13:40
- 事件:session open(代表通過驗證)
- 嘗試登入帳號:test2
- 使用者id:1003

3. 輸入 `sudo cat /var/log/auth.log | grep "COMMAND="`

```
2026-04-22T16:29:46.318034+08:00 marrow-VirtualBox sudo
:    test2 : command not allowed ; TTY=pts/1 ; PWD=/hom
e/test2 ; USER=root ; COMMAND=list
```

- 執行帳號:test2
- 訊息:command not allowed
- 所在位置:/home/test2
- 目標身分:USER=root
- 執行的指令: `list`

```
2026-04-22T16:53:27.878256+08:00 marrow-VirtualBox sudo
:    test2 : command not allowed ; TTY=pts/1 ; PWD=/hom
e/test2 ; USER=root ; COMMAND=/usr/bin/find . -exec /bi
n/sh ; -quit
```

- 執行帳號:test2
- 訊息:command not allowed
- 所在位置:/home/test2

- 目標身分:USER=root
- 執行的指令: `/usr/bin/find . -exec /bin/sh; -quit`

```
2026-04-22T16:57:42.311030+08:00 marrow-VirtualBox sudo
:    test2 : TTY=pts/1 ; PWD=/home/test2 ; USER=root ;C
OMMAND=/usr/bin/apt update
```

- 執行帳號:test2
- 所在位置:/home/test2
- 目標身分:USER=root
- 執行的指令: `/usr/bin/apt update`

第二題

- 作業目標:
  - 學習並掌握 auditd（Linux Audit Daemon）的基本操作方式，透過設定稽核規則來監控重要系統檔案是否被存取或修改。
    本次作業的監控目標檔案包括：
  - /etc/shadow（儲存使用者密碼雜湊的檔案）
  - ~/.bashrc（使用者的 shell 啟動腳本）
  - root 使用者的 crontab 設定檔

參考做法

1. 透過以下指令來更新 apt 並安裝 auditd

```
sudo apt update
sudo apt install auditd -y
```

```
┌──(sharon㉿kali)-[~/homwork]
└─$ sudo apt install auditd -y
The following packages were automatically installed and are no longer required:
  attr                              librdmacm1                        python3-log-symbols
  base58                            librpmsign9                       python3-markdown
  cpp-13                            librtlsdr0                        python3-marshmallow
  cython3                           librtlsdr2                        python3-marshmallow-sqlalchemy
  debtags                           libsoundtouch1                    python3-memcache
  dnsmap                            libsrtp2-1                        python3-mistune0
  faraday-agent-dispatcher          libstrongswan                     python3-mitmproxy-rs
  figlet                            libstrongswan-standard-plugins    python3-mitmproxy-wireguard
  finger                            libsuperlu6                       python3-mnemonic
  gir1.2-girepository-2.0           libtbb12                          python3-multidict
  graphicsmagick                    libtbbbind-2-5                    python3-nassl
  graphicsmagick-imagemagick-compat libtbbmalloc2                     python3-nplusone
  greenbone-feed-sync               libtirpc-dev                      python3-numba
  gvmd                              libubertooth1                     python3-numexpr
  gvmd-common                       libucl1                           python3-odf
  ibverbs-providers                 libvo-aacenc0                     python3-ordered-set
  icu-devtools                      libvo-amrwbenc0                   python3-paho-mqtt
  imagemagick-6-common              libwildmidi2                      python3-pandas
  kali-debtags                      libwmflite-0.2-7                  python3-pandas-lib
  kismet-capture-common             libxm4                            python3-pathspec
  kismet-capture-hak5-wifi-coconut  libxsimd-dev                      python3-pendulum
  kismet-capture-linux-bluetooth    libyajl2                          python3-pgspecial
  kismet-capture-linux-wifi         libzbar0                          python3-pickleshare
  kismet-capture-nrf-51822          libzxing2                         python3-plaster
```

跑完安裝指令之後輸入以下指令來確認 auditd 目前狀況

```
sudo systemctl status auditd
```

從截圖中可以看到，auditd 目前的 Active 狀態為 inactive (dead)，因此接下來需要啟動 auditd 服務。

```
┌──(sharon㉿kali)-[~/homwork]
└─$ sudo systemctl status auditd
[sudo] password for sharon:
○ auditd.service - Security Audit Logging Service
     Loaded: loaded (/usr/lib/systemd/system/auditd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:auditd(8)
             https://github.com/linux-audit/audit-documentation
```

輸入以下指令可以立刻啟動服務，再利用 `status auditd` 來確認是否有啟動成功，若啟動成功，會看到如圖中紫色標示的 active (running)。

```
sudo systemctl start auditd
```

```
┌──(sharon㉿kali)-[~/homwork]
└─$ sudo systemctl start auditd

┌──(sharon㉿kali)-[~/homwork]
└─$ sudo systemctl status auditd
● auditd.service - Security Audit Logging Service
     Loaded: loaded (/usr/lib/systemd/system/auditd.service; disabled; preset: disabled)
     Active: active (running) since Mon 2026-04-20 11:57:01 EDT; 2s ago
       Docs: man:auditd(8)
             https://github.com/linux-audit/audit-documentation
    Process: 2922291 ExecStart=/usr/sbin/auditd (code=exited, status=0/SUCCESS)
   Main PID: 2922292 (auditd)
      Tasks: 2 (limit: 9436)
     Memory: 616.0K (peak: 1.1M)
        CPU: 2ms
     CGroup: /system.slice/auditd.service
             └─2922292 /usr/sbin/auditd

Apr 20 11:57:01 kali systemd[1]: Starting auditd.service - Security Audit Logging Service...
Apr 20 11:57:01 kali systemd[1]: Started auditd.service - Security Audit Logging Service.
Apr 20 11:57:01 kali auditd[2922292]: No plugins found, not dispatching events
Apr 20 11:57:01 kali auditd[2922292]: Init complete, auditd 4.1.2 listening for events (startup state enable)
```

2. 撰寫監控檔案的 rule

```
# 監控 /etc/shadow
sudo auditctl -w /etc/shadow -p rwa -k watch_shadow

# 監控 ~/.bashrc
sudo auditctl -w /root/.bashrc -p rwa -k watch_bashrc

# 監控 root 的 crontab
sudo auditctl -w /var/spool/cron/crontabs/root -p rwa -k watch_crontab
```

指令參數說明：

- `-w` ：watch，指定要監控的檔案路徑
- `-p rwa` ：監控 read / write / attribute change，也就是讀取、寫入與屬性變更。
- `-k` ：設定一個關鍵字 key，方便之後搜尋記錄

執行後若出現 `Old style watch rules are slower` 是正常的警告訊息，不影響功能，規則依然有效。

```
┌──(sharon㉿kali)-[~/homwork]
└─$ sudo auditctl -w /etc/shadow -p rwa -k watch_shadow
Old style watch rules are slower

┌──(sharon㉿kali)-[~/homwork]
└─$ sudo auditctl -w /root/.bashrc -p rwa -k watch_bashrc
Old style watch rules are slower

┌──(sharon㉿kali)-[~/homwork]
└─$ sudo auditctl -w /var/spool/cron/crontabs/root -p rwa -k watch_crontab
Old style watch rules are slower
```

3. 使用 sudo auditctl -l 指令列出目前所有已設定的監控規則

新增規則後，使用 `sudo auditctl -l` 指令來確認規則是否已成功套用，若有成功從輸出結果可以看到三條規則都已正確載入：

- `/etc/shadow` → 以 `watch_shadow` 為關鍵字監控
- `/root/.bashrc` → 以 `watch_bashrc` 為關鍵字監控
- `/var/spool/cron/crontabs/root` → 以 `watch_crontab` 為關鍵字監控

這代表 auditd 已開始監控這三個檔案，任何存取行為都會被記錄下來。

```
┌──(sharon㉿kali)-[~/homwork]
└─$ sudo auditctl -l
-w /etc/shadow -p rwa -k watch_shadow
-w /root/.bashrc -p rwa -k watch_bashrc
-w /var/spool/cron/crontabs/root -p rwa -k watch_crontab
```

4. 嘗試修改或讀取上述監控的檔案，然後使用 `sudo ausearch -k watch_crontab` 指令查詢是否有記錄產生：

1. **觸發監控事件**

   以 root 權限嘗試讀取或修改您設定的監控目標 ：

- 讀取密碼雜湊檔： 執行 `sudo cat /etc/shadow` 。

- 讀取 root 的 bashrc： 執行 `sudo cat /root/.bashrc` 。

```
┌──(david㉿kali)-[~]
└─$ sudo cat /root/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
```

- 存取 crontab 檔案： 執行 `sudo ls /var/spool/cron/crontabs/root` 。

```
┌──(david㉿kali)-[~]
└─$ sudo ls /var/spool/cron/crontabs/root
ls: 無法存取 '/var/spool/cron/crontabs/root': 沒有此一檔案或目錄
```

「有監控規則」並不等於「該檔案一定存在」

原因:

該功能尚未被使用（預設不存在）：

在許多 Linux 發行版中，如果 root 使用者從未執行過 crontab -e 來設定排程任務，系統就不會建立 /var/spool/cron/crontabs/root 這個檔案 。

規則是「路徑導向」而非「實體導向」：

稽核規則只是在核心中掛載一個「監視器」在該路徑上 。即使路徑是空的，規則依然會存在於 auditctl -l 的清單中 。

權限與存在性的差異：

ls 指令會檢查檔案系統的真實狀態，而 auditctl 只是紀錄你的意圖。當你執行 ls 時，系統發現實體檔案不存在，因此回報錯誤。

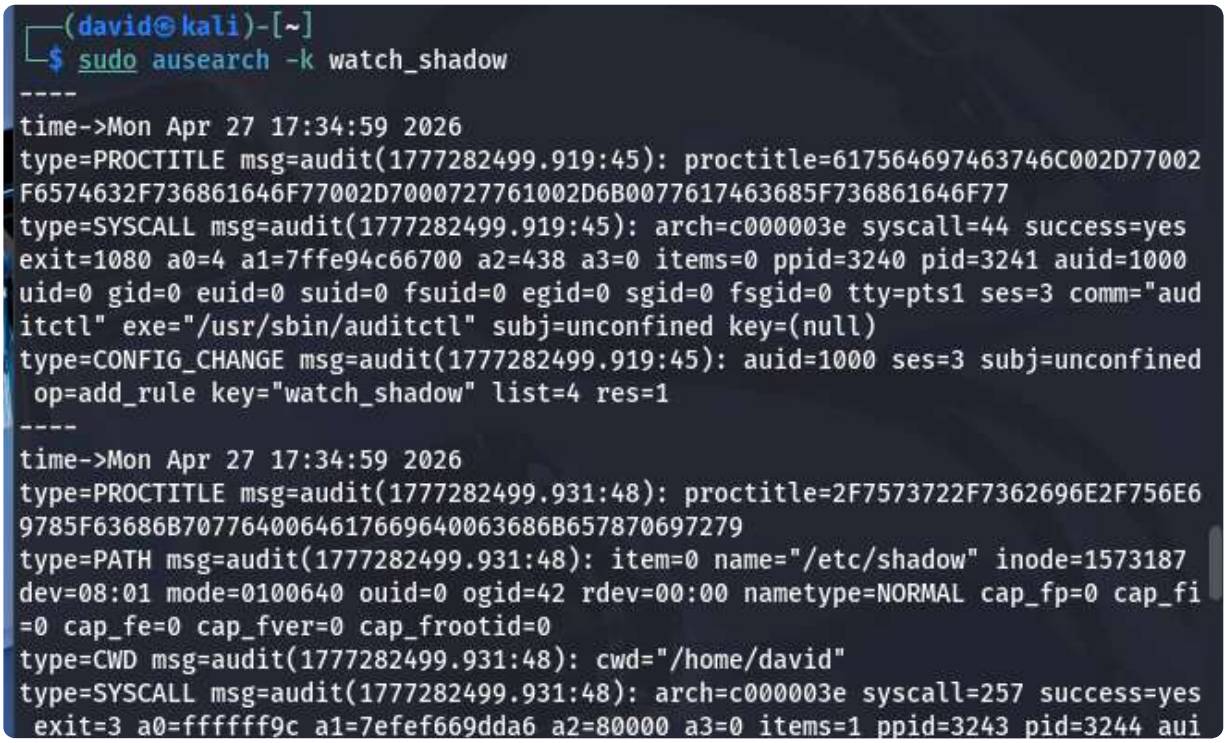
所以需要做的應該是觸發一個「確定存在」的檔案。

2. 使用 ausearch 查詢紀錄

   執行完上述動作後，使用您在規則中設定的「關鍵字（Key）」來搜尋特定的日誌 ：

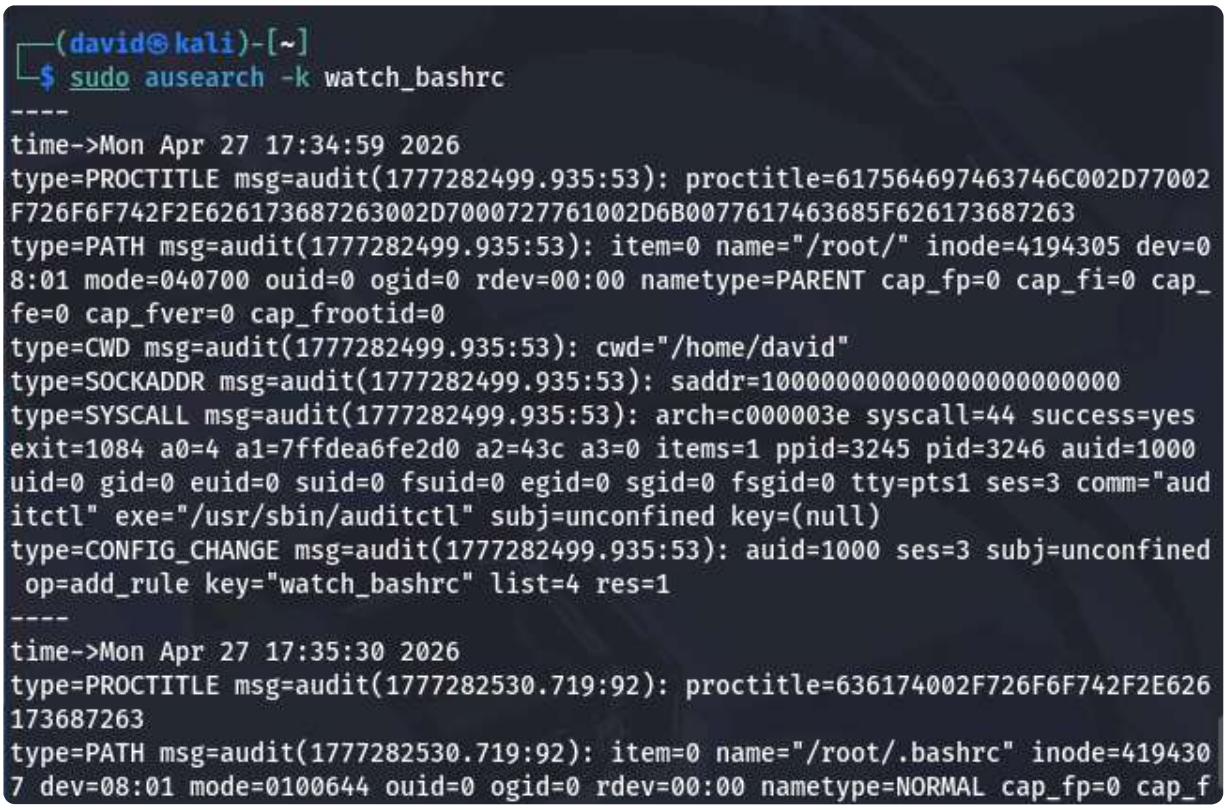
- 查詢 shadow 檔案存取紀錄：

```
sudo ausearch -k watch_shadow
```



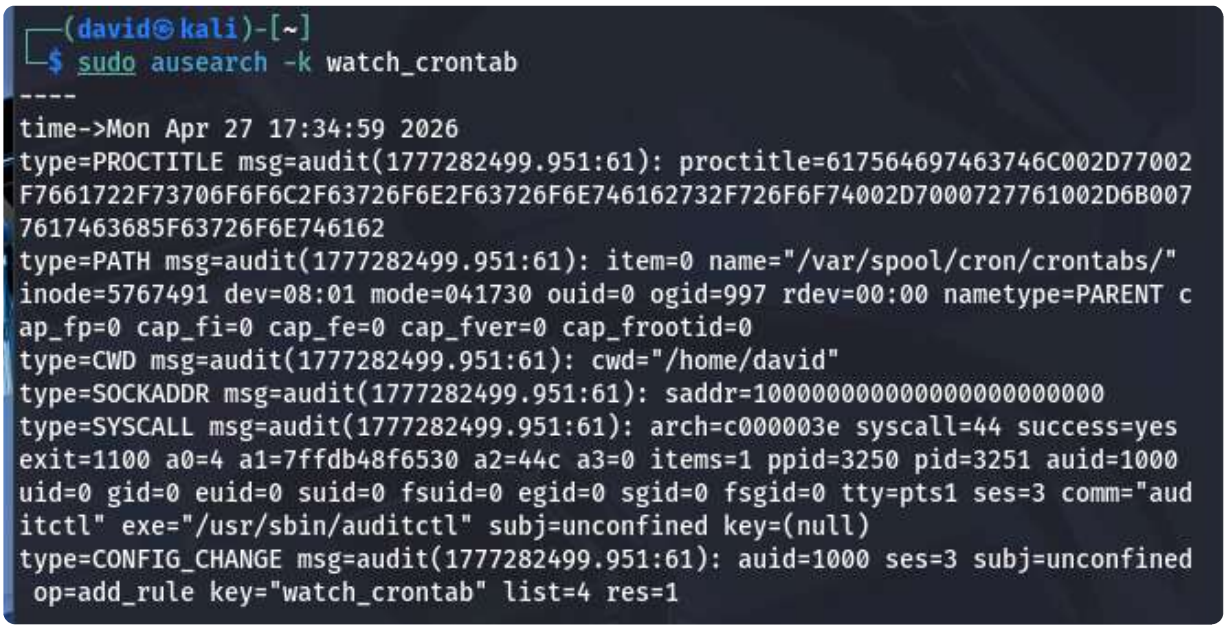
- 查詢 bashrc 檔案存取紀錄：

```
sudo ausearch -k watch_bashrc
```



- 查詢 crontab 檔案存取紀錄：

```
sudo ausearch -k watch_crontab
```



3. 觀察輸出結果

   在 ausearch 的輸出中，您可以確認以下資訊：

- time： 事件發生的精確時間。
- name： 被存取的檔案路徑（如 /etc/shadow）。
- exe： 執行該動作的程式路徑（如 /usr/bin/cat）。
- auid： 原始登入使用者的 ID（即使使用了 sudo，也能追蹤到是哪個使用者操作的）。
- res： 操作結果（success 或 failed）。

結論

1. 系統日誌 (/var/log/auth.log) 的重要性身份驗證追蹤： 透過觀察 auth.log，管理員可以清楚辨識正常的登入行為與惡意的暴力破解嘗試（如 authentication failure 紀錄）。權限變更稽核： sudo 指令的紀錄能確保所有具備管理權限的操作都有跡可循，避免使用者越權執行未經授權的指令 。
2. 稽核守護進程 (auditd) 的防禦價值主動監控： 與被動記錄登入資訊不同，auditd 允許針對「特定敏感檔案」設定規則，一旦檔案被存取或修改（rwa），系統會立即產生詳細紀錄 。合規與鑑識： 監控 /etc/shadow 或 crontab 等關鍵配置，可以有效偵測未經授權的後門植入（如修改啟動腳本或排程任務），是系統安全稽核與事後數位鑑識的重要工具 。