Opgave 27 - Audit af en fil for ændringer

Formålet med øvelsen

Formålet med denne øvelse er at **give en introduktion til, hvordan audit-regler kan konfigureres med auditd**.
Auditd giver mulighed for at overvåge kritiske systemfiler og registrere ændringer, som kan være relevante for sikkerhed og compliance.
I denne øvelse lærer du at opsætte audit-regler både midlertidigt og permanent samt at analysere de loggede hændelser.

Information

For at definere begivenheder, der udløser en *audit log*, skal man opsætte en auditregel. Dette kan gøres på to måder:
1. **Dynamisk** ved at tilføje reglen via værktøjet `auditctl`.
2. **Permanent** ved at tilføje reglen i konfigurationsfilen `/etc/audit/rules.d/`.

I denne øvelse lærer du at opsætte begge typer regler og analysere audit-logs.

Instruktioner

Brug af generative AI'er i undervisningen

Følgende tilgang må kun bruges i undervisningen (Aldrig i en virksomhed).

Generative AI'er, såsom ChatGPT, kan være nyttige til at analysere logfiler i en læringskontekst. Hvis du er i tvivl om outputtet af en audit-log, kan du prøve at bruge en generativ AI til at få en forklaring.

Eksempel på en forespørgsel til en AI:

```
Jeg brugte denne kommando: <Kommando>.
Jeg fik dette svar: <Log output>.
Kan du forklare, hvad dette betyder?
```

 Vigtigt:
I en **virksomhedsmæssig kontekst** må du **aldrig** indsætte interne logfiler eller følsomme data i en generativ AI, da dette kan bryde virksomhedens sikkerhedspolitikker og føre til datalæk. Brug derfor kun denne metode i læringsøjemed her på uddannelsen.

Tilføj auditregel med `auditctl`

1. Opret en auditregel, der overvåger ændringer og skrivninger til filen `/etc/passwd`, med kommandoen:

```
auditctl -w /etc/passwd -p wa -k user_change
```

Forklaring:
- `-w` står for "where" og angiver den fil, der skal overvåges.
- `-p` står for "permissions" og definerer, hvilke handlinger der skal overvåges (`wa` = write + attribute ændringer).
- `-k` tildeler en nøgle (*key*), der gør det lettere at søge efter relaterede hændelser.

1. Udskriv en rapport over loggede hændelser med:

```
aureport -i -k | grep user_change
```

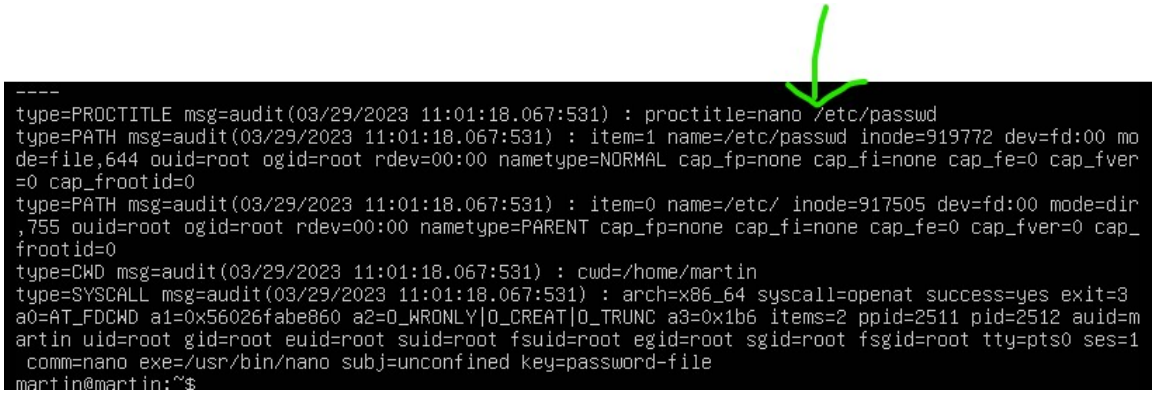
Forklaring:
- `-i` betyder "interpret" og oversætter numeriske værdier (fx bruger-ID) til læsbare navne.
- `-k` filtrerer loggen baseret på den tidligere definerede nøgle.

Rapporten viser hændelser i formatet: *Dato/tid, key, succes-status, udførende proces (bruger, process-id osv.)*.

1. Opret en ny bruger med `useradd`. Dette vil tilføje en ny linje i `/etc/passwd`, da brugeroplysninger gemmes her.
2. Udskriv loggen igen med `aureport -i -k | grep user_change` og verificér, at der nu er to nye rækker i rapporten (én for skrivning til filen og én for ændring af metadata).
3. Brug `ausearch` til at analysere hændelserne mere detaljeret:

```
ausearch -i -k user_change
```

Dette giver en mere detaljeret logvisning, men kan også være mere uoverskueligt.



Tilføj auditregel med konfigurationsfilen

Auditctl-regler er **ikke persistente**, hvilket betyder, at de forsvinder ved en genstart. For at gøre reglerne permanente, skal de gemmes i `/etc/audit/rules.d/`.

1. Åbn filen `/etc/audit/audit.rules`. Den bør se ud som vist nedenfor:

```
## This file is automatically generated from /etc/audit/rules.d
-D
-b 8192
-f 1
--backlog_wait_time 60000
```

Denne fil bliver indlæst ved opstart af auditd og indeholder alle aktive audit-regler.
Bemærk: Øverst i filen står der, at den autogenereres ud fra `/etc/audit/rules.d/`.

1. Opret en ny fil med den aktuelle auditregel ved at køre:

```
sh -c "auditctl -l > /etc/audit/rules.d/custom.rules"
```

Vigtigt! Reglen fra den tidligere sektion skal stadig være aktiv. Kontrollér dette med `auditctl -l` inden du fortsætter.

1. Genstart auditd med:

```
systemctl restart auditd
```

2. Udskriv indholdet af `/etc/audit/audit.rules`, som nu bør indeholde den nye regel:

```
martin@martin:~$ sudo cat  /etc/audit/audit.rules
## This file is automatically generated from /etc/audit/rules.d
-D
-b 8192
-f 1
--backlog_wait_time 60000
-w /etc/passwd -p wa -k user_change
```

3. Tilføj en ny bruger med `useradd`, og verificér, at reglen fungerer ved at kontrollere, om ændringer i `/etc/passwd` registreres.
4. **Slet filen `/etc/audit/rules.d/custom.rules` efter øvelsen** og genstart auditd.

Overvågning af en tekstfil

Auditd kan også bruges til at overvåge ændringer i almindelige tekstfiler. Dette kan være nyttigt for at sikre **filintegritet** og spore, hvem der har foretaget ændringer.

1. Opret en fil med noget tekst:

```
echo "Dette er en testfil" > testfil.txt
```

2. Konfigurer Auditd til at overvåge filen:

```
auditctl -w /path/to/testfil.txt -p wa -k textfile_watch
```

3. Tilføj noget tekst til filen:

```
echo "Ny linje tilføjet" >> testfil.txt
```

4. Verificér, at ændringen er blevet registreret i audit-loggen:

```
ausearch -i -k textfile_watch
```

Refleksioner

Overvej følgende spørgsmål efter at have gennemført øvelsen:

- Hvordan kan auditd hjælpe med at opdage uautoriserede ændringer i systemfiler?
- Hvilke typer af filer eller systemområder ville være kritiske at overvåge i en produktionsserver?
- Hvordan kan AuditD integreres med F.eks. Wazuh?

Auditd vs. Sysmon

Hvis du har arbejdet med sikkerhedslogning på **Windows**, kender du måske **Sysmon**. Det fungerer på samme måde som auditd, men er designet til Microsoft-systemer.

Funktion	Auditd (Linux)	Sysmon (Windows)
Platform	Linux	Windows
Overvåger filer?	Ja, filændringer og adgang	Ja, filændringer
Overvåger processer?	Ja, systemkald og processkabelse	Ja, detaljeret procesovervågning
Overvåger netværk?	Begrænset, kræver ekstra moduler	Ja, registrerer netværksforbindelser
Logdestination	`/var/log/audit/audit.log`	Windows Event Log
SIEM-integration	Ja (ELK, Splunk, Wazuh)	Ja (Splunk, ELK, Defender ATP)

2025-04-03 05:55:59