

SSO Práctica 1 Sesión 3

Andrés Merlo Trujillo

Ejercicio 1

Con la orden `aa-status` o la orden `apparmor_status` se pueden ver los perfiles activos en Ubuntu:

Ahora voy a elegir el perfil `/usr/bin/freshclam`, para poder ver el archivo del perfil asociado basta con irse al directorio `/etc/apparmor.d` y el archivo se denomina igual que la ruta absoluta del mismo, pero en vez de usar “/” se utilizan puntos. Por tanto, el archivo deseado es: `/etc/apparmor.d/usr.bin.freshclam`.

```
File: usr.bin.freshclam
1 # vim:syntax=apparmor
2 # Author: Jamie Strandboge <jamie@ubuntu.com>
3 # Last Modified: Sun Aug  3 09:39:03 2008
4
5 #include <tunables/global>
6
7 /usr/bin/freshclam flags=(attach_disconnected) {
8   #include <abstractions/base>
9   #include <abstractions/nameservice>
10  #include <abstractions/user-tmp>
11  #include <abstractions/openssl>
12
13  capability dac_override,
14  capability chown,
15
16  capability setgid,
17  capability setuid,
18
19  @{PROC}/filesystems r,
20  owner @{PROC}/[0-9]*/status r,
21
22  /etc/clamav/clamd.conf r,
23  /etc/clamav/freshclam.conf r,
```

```
23  /etc/clamav/freshclam.conf r,
24  /etc/clamav/onerrorexecute.d/* mr,
25  /etc/clamav/onupdateexecute.d/* mr,
26  /etc/clamav/virusevent.d/* mr,
27
28  owner @{HOME}/.clamtk/db/ rw,
29  owner @{HOME}/.clamtk/db/** rwk,
30
31  owner @{HOME}/.clamav/database/ rw,
32  owner @{HOME}/.clamav/database/** rwk,
33
34  /usr/bin/freshclam mr,
35
36  /var/lib/clamav/ r,
37  /var/lib/clamav/** krw,
38
39  /var/log/clamav/* krw,
40  /{,var}/run/clamav/freshclam.pid w,
41  /{,var}/run/clamav/clamdctl rw,
42
43  deny /{,var}/run/samba/{gencache,unexpected}.tdb mrwkl,
44
45  # Site-specific additions and overrides. See local/README for details.
46  #include <local/usr.bin.freshclam>
47 }
```

(END)

Las componentes principales son las siguientes:

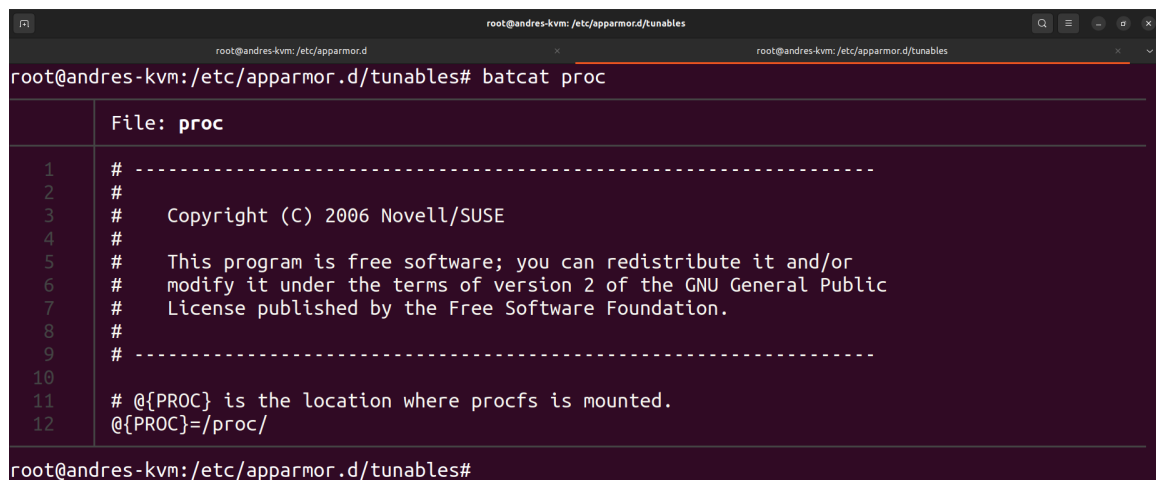
- `#include <tunables/global>` Carga un archivo que contiene las definiciones de las variables.
- `/usr/bin/freshclam` Ruta absoluta del binario.
- `#include <abstractions/base>` Obtiene los componentes de los perfiles de AppArmor para simplificar el desarrollo de perfiles.
- `#include <abstractions/nameservice>` Incluye las reglas para permitir DNS, LDAP, NIS, SMB, contraseñas de usuarios y grupos, servicios y “lookups” de protocolos
- `#include <abstractions/user-tmp>` Permite acceder a los directorios temporales
- `#include <abstractions/openssl>` Permite acceder a los archivos correspondientes a OpenSSL.
- `{,var/}` Permite eliminar líneas innecesarias, poniendo los directorios similares dentro de la lista entre llaves.

En este caso las opciones son `/run/clamav/freshclam.pid` y `/var/run/clamav/freshclam.pid`

- `capability ...` Indica las capabilities que tiene permitidas hacer en el sistema. El listado de todas ellas se puede ver usando `man 7 capabilities`.
- `owner archivo` Indica que solo puede acceder al archivo indicado si es el propietario del mismo.
- `deny archivo` Deniega el acceso al archivo indicado.

Además, aparecen variables del tipo “@...”.

El valor de estas variables se almacenan en `/etc/apparmor.d/tunables/file`, donde file es el nombre de la variable.



```
root@andres-kvm:/etc/apparmor.d/tunables# batcat proc
File: proc
1 # -----
2 #
3 #   Copyright (C) 2006 Novell/SUSE
4 #
5 #   This program is free software; you can redistribute it and/or
6 #   modify it under the terms of version 2 of the GNU General Public
7 #   License published by the Free Software Foundation.
8 #
9 # -----
10
11 # @{PROC} is the location where procfs is mounted.
12 @{PROC}=/proc/
root@andres-kvm:/etc/apparmor.d/tunables#
```

Figura 1: Ejemplo de archivo usado por las variables, en este caso de PROC.

Las que aparecen en este perfil son:

- `@{HOME}`: Lista de todos los `home` de los usuarios, incluido el root.
- `@{PROC}`: Directorio donde procfs es montado.

También se puede ver que contiene una lista de archivos y directorios junto con sus permisos, estos son los archivos o directorios a los que puede tener acceso, determinado por los switches que se muestran a continuación:

- **r**: Modo lectura.
- **w**: Modo escritura.
- **a**: Modo adjuntar (append).
- **k**: Modo de bloqueo de archivo.
- **l**: Modo de enlace.
- **ux**: Modo de ejecución sin restricciones.
- **Ux**: Modo de ejecución sin restricciones. Además, limpia el entorno (scrub the environment).
- **px**: Ejecución discreta del perfil.
- **Px**: Modo de ejecución discreta del perfil. Además, limpia el entorno (scrub the environment).
- **ix**: Modo de ejecución heredada.
- **m**: Permite PROT_EXEC con llamadas a `mmap`.
- **Cx**: Permite transiciones a un perfil hijo. Con la C mayúscula se usa “secure exec” de glibc.

Ejercicio 2

Voy a generar un perfil para el programa `nano`, la característica principal que voy a añadir es prohibirle el acceso a un archivo denominado `/root/archivoProhibido` el cual contiene lo siguiente:

```
root@andres-kvm:~# batcat archivoProhibido
```

	File: archivoProhibido
1	Este archivo no se puede modificar

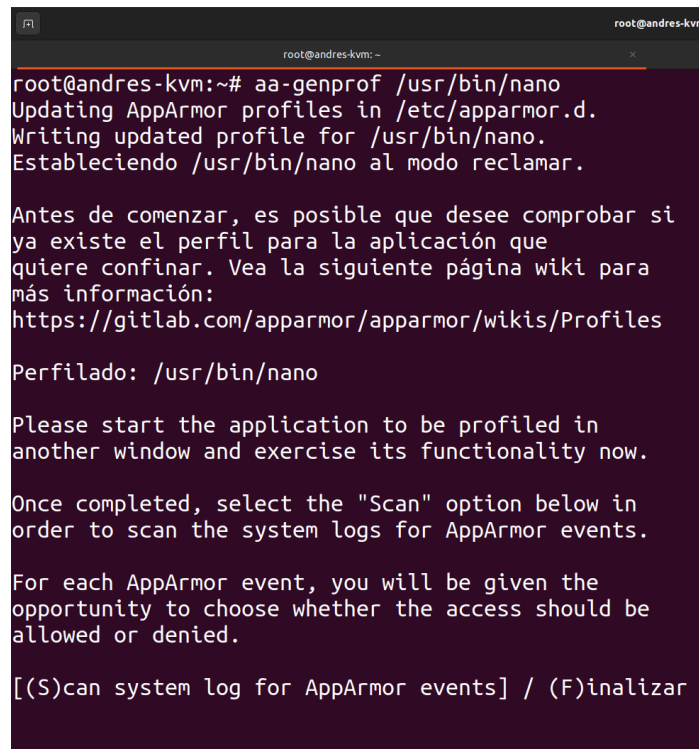
```
root@andres-kvm:~# █
```

Figura 2: Contenido de “archivoProhibido”.

Para saber su ruta absoluta se puede usar la orden `which nano`:

```
root@andres-kvm:~# which nano
/usr/bin/nano
root@andres-kvm:~#
```

Ahora para generar el perfil se ejecuta el comando `aa-genprof /usr/bin/nano`:



```
root@andres-kvm:~# aa-genprof /usr/bin/nano
Updating AppArmor profiles in /etc/apparmor.d.
Writing updated profile for /usr/bin/nano.
Estableciendo /usr/bin/nano al modo reclamar.

Antes de comenzar, es posible que desee comprobar si
ya existe el perfil para la aplicación que
quiere confinar. Vea la siguiente página wiki para
más información:
https://gitlab.com/apparmor/apparmor/wikis/Profiles

Perfilado: /usr/bin/nano

Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" option below in
order to scan the system logs for AppArmor events.

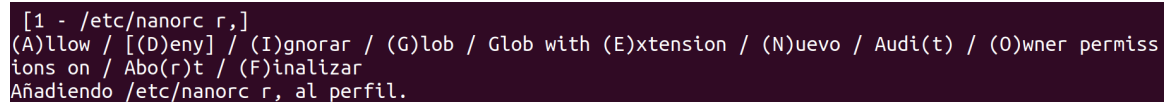
For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

[(S)can system log for AppArmor events] / (F)inalizar
```

Figura 3: Contenido de “archivoProhibido”.

Ahora pide que abramos el programa a perfilar y pulsemos en el botón de escanear. Abriendo `nano` en otra terminal permitirá continuar con el proceso.

A continuación aparecerán distintos archivos y capabilities relacionadas con las que debemos dar acceso o no.



```
[1 - /etc/nanorc r,]
(A)llow / [(D)eny] / (I)gnorar / (G)lob / Glob with (E)xtension / (N)uevo / Audi(t) / (O)wner permiss
ions on / Abo(r)t / (F)inalizar
Añadiendo /etc/nanorc r, al perfil.
```

Figura 4: Al pulsar la tecla “O” desaparece la palabra “owner”, permitiendo acceso a todos.

Al ser el archivo de configuración de `nano`, es recomendable deshabilitar los permisos de propietario, para que los demás usuarios puedan usarlo y permitirlo.

Además, aparece la opción de denegar el acceso a `/etc/passwd`, tras varias modificaciones he llegado a la conclusión de que es necesario para que detecte los usuarios que no sean root, por lo que hay que ponerle el mismo ajuste que a `/etc/nanorc`.

Finalmente, el archivo generado por defecto es el siguiente:

```
root@andres-kvm:~# batcat /etc/apparmor.d/usr.bin.nano

File: /etc/apparmor.d/usr.bin.nano

1 # Last Modified: Tue Oct 18 17:20:41 2022
2 abi <abi/3.0>,
3
4 include <tunables/global>
5
6 /usr/bin/nano {
7     include <abstractions/base>
8     include <abstractions/evince>
9
10    /etc/nanorc r,
11    /etc/passwd r,
12    /usr/bin/nano mr,
13    owner /etc/nsswitch.conf r,
14    owner /root/.local/share/nano/search_history r,
15
16 }
```

Esta configuración va a prohibir por defecto el acceso a todos los directorios, salvo los explícitamente mencionados. Si se quiere que se permita acceso a los directorios `/home` y `/root`, pero prohibiendo el acceso a `/root/archivoProhibido` se debe poner lo siguiente:

```
root@andres-kvm:~# batcat /etc/apparmor.d/usr.bin.nano

File: /etc/apparmor.d/usr.bin.nano

1 # Last Modified: Tue Oct 18 17:20:41 2022
2 abi <abi/3.0>,
3
4 include <tunables/global>
5
6 /usr/bin/nano {
7     include <abstractions/base>
8     include <abstractions/evince>
9
10    /etc/nanorc r,
11    /etc/passwd r,
12    /usr/bin/nano mr,
13    owner /etc/nsswitch.conf r,
14    owner /root/.local/share/nano/search_history r,
15
16    /root/* rw,
17    /home/* rw,
18    /home/andres/* rw,
19    deny /root/archivoProhibido rw,
20 }
```

Ahora, haciendo `systemctl reload apparmor` se recargan todos los perfiles y como se puede observar, si hago `nano /root/prueba` permite la creación del archivo.

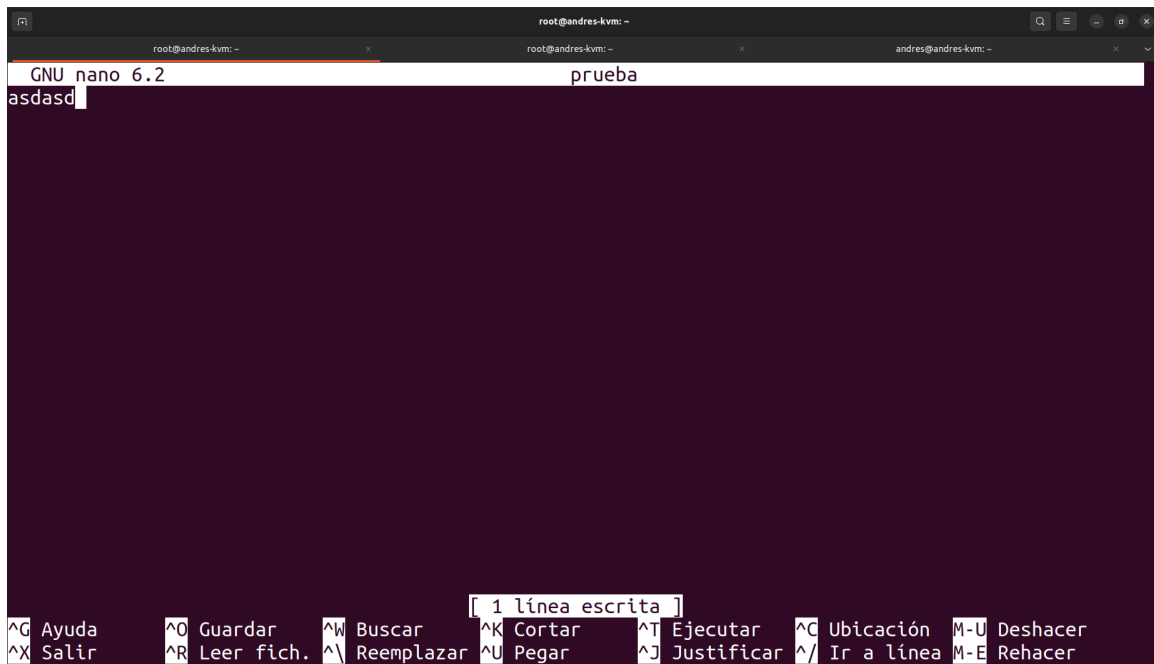


Figura 5: Permite la creación de un archivo en el directorio home de root.

Sin embargo, si intento hacer `nano /root/archivoProhibido` no permite ni visualizarlo:

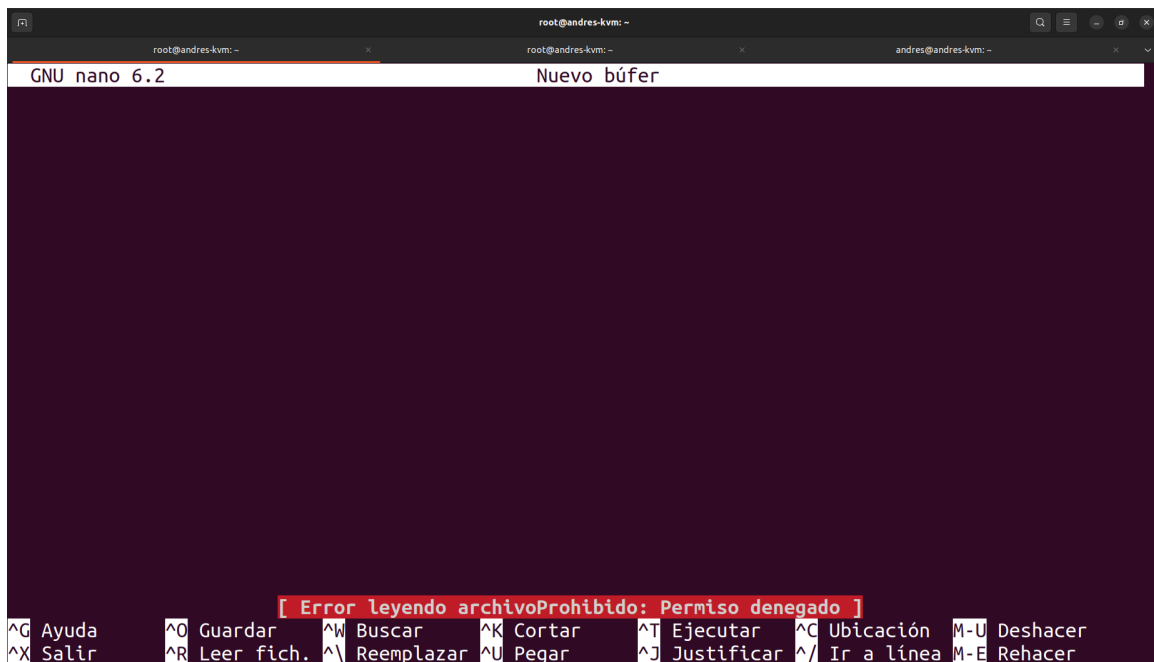


Figura 6: Sigue prohibiendo el acceso a este archivo, ya que está explícitamente prohibido con “deny”.