Ejercicio 41.2 Explore la seguridad de **apparmor**

- Este ejercicio solo se puede realizar en un sistema (como **Ubuntu**) donde está instalado **AppArmor**.
- Lo siguiente se probó en **Ubuntu 17.04**, pero debería funcionar en otros sistemas compatibles con **AppArmor**, como **OpenSUSE**, donde los comandos **apt-get** deberían reemplazarse por **zypper**.

En **Ubuntu**, la utilidad **/bin/ping** se ejecuta con SUID habilitado. Para este ejercicio, copiaremos **ping** a **ping-x** y ajustaremos las capacidades para que el programa funcione.

Luego, crearemos un perfil de **AppArmor**, instalaremos y verificaremos que nada haya cambiado. La modificación del perfil de **AppArmor** y la adición de capacidades le permitirán al programa más funcionalidad.

1. Asegúrese de que todos los paquetes necesarios estén instalados:

```
student@ubuntu:~$ sudo apt-get install apparm*
```

2. Cree una copia de **ping** (llamada **ping-x**) y verifique que no tenga permisos o capacidades especiales iniciales. Además, no puede funcionar cuando lo ejecuta por **student**, un usuario normal:

```
student@ubuntu:~$ sudo cp /bin/ping /bin/ping-x
student@ubuntu:~$ sudo ls -l /bin/ping-x

-rwxr-xr-x 1 root root 64424 Oct 17 10:12 /bin/ping-x

student@ubuntu:~$ sudo getcap /bin/ping-x
student@ubuntu:~$

student@ubuntu:~$ ping-x -c3 -4 127.0.0.1
ping: socket: Operation not permitted
```

3. Establezca las **capacidades** y vuelva a intentar: **ping-x**:

```
student@ubuntu:~$ sudo setcap cap_net_raw+ep  /bin/ping-x

student@ubuntu:~$ ping-x  -c3 -4 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.092 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.093 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.086 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2034ms
rtt min/avg/max/mdev = 0.086/0.090/0.093/0.008 ms
```

   El programa modificado **ping-x** ahora funciona normalmente.

4. Verifique que no haya un perfil de **AppArmor** preexistente para **ping-x**, pero hay un perfil para **ping**. Determine el estado de programa de **ping** actual:

```
student@ubuntu:~$ sudo aa-status
```

   La salida de **aa-status** es larga, por lo que podemos **grep** para las líneas interesantes:

```
student@ubuntu:~$ sudo aa-status | grep -e "^[[:alnum:]]" -e ping

apparmor module is loaded.
87 profiles are loaded.
```



```
51 profiles are in enforce mode.
   ping
36 profiles are in complain mode.
17 processes have profiles defined.
6 processes are in enforce mode.
11 processes are in complain mode.
0 processes are unconfined but have a profile defined.
```

Podemos ver que el **ping** tiene un perfil que está cargado y habilitado para su aplicación.

5. A continuación, construiremos un nuevo perfil para **ping-x**. Este paso requiere dos ventanas terminales.
   La primera ventana (**window1**) ejecutará el comando **aa-genprof**. Esto generará un perfil de **AppArmor** al escanear `/var/log/syslog` para errores de **AppArmor**.
   La segunda ventana (**window2**) se usará para ejecutar **ping-x**. (Consulte la página **man** para **aa-genprof** para obtener información adicional.)
   En **window1**:

```
student@ubuntu:~$ sudo aa-genprof /bin/ping-x
Writing updated profile for /bin/ping-x.
Setting /bin/ping-x to complain mode.

Before you begin, you may wish to check if a
profile already exists for the application you
wish to confine. See the following wiki page for
more information:
http://wiki.apparmor.net/index.php/Profiles

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.

Profiling: /bin/ping-x

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

   En **window2**:

```
student@ubuntu:~$ ping-x  -c3 -4 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.099 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.120 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.114 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2041ms
rtt min/avg/max/mdev = 0.099/0.111/0.120/0.008 ms
```

   En **window1**:
   El comando `ping-x` ha completado, ahora debemos instruir a **aa-genprof** para que busque la información requerida para agregar al perfil. Puede requerir varios **escaneos** para recopilar toda la información para el perfil.
   Ingrese `S` para escanear:

```
Reading log entries from /var/log/syslog.
Updating AppArmor profiles in /etc/apparmor.d.
Complain-mode changes:

Profile:    /bin/ping-x
Capability: net_raw
Severity:   8
```



```
 [1 - capability net_raw,]
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
```

Ingrese `A` para permitir la capacidad:

```
Adding capability net_raw, to profile.

Profile:        /bin/ping-x
Network Family: inet
Socket Type:    raw

 [1 - network inet raw,]
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
```

Ingrese `A` para permitir la familia de la red:

```
Adding network inet raw, to profile.

Profile:        /bin/ping-x
Network Family: inet
Socket Type:    dgram

 [1 - #include <abstractions/nameservice>]
  2 - network inet dgram,
(A)llow / [(D)eny] / (I)gnore / Audi(t) / Abo(r)t / (F)inish
```

Ingrese `A` para agregar el diagrama de tipo de socket al perfil:

```
Adding #include <abstractions/nameservice> to profile.

= Changed Local Profiles =

The following local profiles were changed. Would you like to save them?

 [1 - /bin/ping-x]
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w (C)lean profiles / Abo(r)t
```

Ingrese `S` para guardar el nuevo perfil:

```
Writing updated profile for /bin/ping-x.

Profiling: /bin/ping-x

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

Ingrese `F` para terminar:

```
Setting /bin/ping-x to enforce mode.

Reloaded AppArmor profiles in enforce mode.

               Please consider contributing your new profile!
See the following wiki page for more information:
http://wiki.apparmor.net/index.php/Profiles

Finished generating profile for /bin/ping-x.
```

6. Ver el perfil creado, que se ha almacenado en `/etc/appamor.d/bin.ping-x`.

```
student@ubuntu:~$ sudo cat /etc/apparmor.d/bin.ping-x
# Last Modified: Tue Oct 17 11:30:47 2017
#include <tunables/global>

/bin/ping-x {
  #include <abstractions/base>
  #include <abstractions/nameservice>

  capability net_raw,
```



```
  network inet raw,

  /bin/ping-x mr,
  /lib/x86_64-linux-gnu/ld-*.so mr,

}
```

7. La utilidad **aa-genproc** instala y activa la nueva politica, por lo que debe estar lista para usar, y las políticas se pueden volver a cargar bajo demanda con el comando `systemctl reload apparmor`. Para evitar posibles problemas y verificar que los cambios sobrevivan, reinicie el sistema.

   Una vez que el sistema se ha reiniciado, como usuario `student`, verifique que **ping-x**aún funcione con el nuevo perfil habilitado. Haga ping al localhost por dirección IP:

   ```
   student@ubuntu:~$ ping-x -c3 -4 127.0.0.1
   PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
   64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.057 ms
   64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.043 ms
   64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.095 ms

   --- 127.0.0.1 ping statistics ---
   3 packets transmitted, 3 received, 0% packet loss, time 2027ms
   rtt min/avg/max/mdev = 0.043/0.065/0.095/0.021 ms
   ```

8. Esto debería funcionar como se esperaba. El perfil es muy específico, y **AppArmor** no permitirá la funcionalidad fuera del parámetros especificados. Para verificar que **AppArmor** esté protegiendo esta aplicación, intente hacer ping a la dirección localhost **IPV6**.

   Esto debería fallar:

   ```
   student@ubuntu:~$ ping-x -c3  -6 ::1
   ping: socket: Permission denied
   ```

   (Tenga en cuenta, el opción `-6` significa usar solo **IPv6** y `::1` es el anfitrión local en **IPv6**.)

   La salida indica que hay un problema con el socket. Si se examina el registro del sistema, se descubrirá que nuestro programa **ping-x** no tiene acceso a **IPv6** dentro de **AppArmor**:

   ```
   766:104): apparmor="DENIED" operation="create" profile="/bin/ping-x"
   pid=2709 comm="ping-x" family="inet6" sock_type="raw" protocol=58
   requested_mask="create" denied_mask="create
   ```

9. Para corregir esta deficiencia, vuelva a ejecutar **aa-genprof** como lo hicimos anteriormente, y en **window2**, haga ping al **IPv6** loopback y anexe las opciones adicionales.

