Seguridad

Explicación de práctica 5 - parte 2

Sistemas Operativos

Facultad de Informática
Universidad Nacional de La Plata

2025

DAC vs MAC

- Discretionary Access Control (DAC)
  - Matriz de acceso
  - Permisos para un determinado usuario y grupo a un recurso.
  - Concepto de ownership: El/la owner puede cambiar los permisos de un recurso.
  - Ej: UGO en archivos
- Mandatory Access Control (MAC)
  - Se basa en policies que los usuarios no pueden modificar.
  - Primero se verifica la matriz de acceso.
  - Luego se verifica que las policies permitan la acción.
  - Ej: Linux Security Modules: SELinux, AppArmor, Tomoyo, Lockdown, etc…

AppArmor

- Es una implementación Linux Security Module.
- Más sencillo que SELinux.
- Actualmente mantenido por Canonical.
- Provee un conjunto de profiles pero es fácil crear nuevos.
- Se basa en los paths a los ejecutables para determinar qué profile aplicar.
- Como todo MAC complementa a la matriz de acceso tradicional.

AppArmor

- Es una implementación Linux Security Module.
- Más sencillo que SELinux.
- Actualmente mantenido por Canonical.
- Provee un conjunto de profiles pero es fácil crear nuevos.
- Se basa en los paths a los ejecutables para determinar qué profile aplicar.
- Como todo MAC complementa a la matriz de acceso tradicional.

AppArmor

- **AppArmor** es una solución de seguridad para aplicaciones diseñada específicamente para aplicar confinamiento de privilegios a procesos.
- Permite al administrador especificar el dominio de actividades que un programa puede realizar mediante el desarrollo de un perfil de seguridad.
- Un perfil de seguridad es una lista de archivos a los que el programa puede acceder y las operaciones que puede realizar.
- Los perfiles de AppArmor pueden operar en dos modos:
  - **enforcing**: aplica estrictamente las reglas del perfil y bloquea cualquier acción no autorizada.
  - **complain**: no bloquea las acciones no autorizadas, pero las loguea, lo que permite al administrador ajustar los perfiles sin interrumpir el funcionamiento del programa.

AppArmor herramientas

- aa-enabled: Retorna exit status 0 si el kernel lo soporta
- aa-status: Perfiles
- aa-unconfined: Lista programas que tengan sockets abiertos detallando si tienen un profile asociado
- aa-complain <path-programa>
- aa-enforce <path-programa>
- aa-genprof <path-programa>

AppArmor herramientas

- aa-enabled: Retorna exit status 0 si el kernel lo soporta
- aa-status: Perfiles
- aa-unconfined: Lista programas que tengan sockets abiertos detallando si tienen un profile asociado
- aa-complain <path-programa>: Modo comply
- aa-enforce <path-programa>: Modo enforce
- aa-genprof <path-programa>: Asistente para generar profile
- apparmor_parse -r <path-profile>: Carga el profile (por ej. si fue modificado)
- journal -xf
- tail -f /var/log/audit/audit.log

Fuentes

- https://apparmor.net/
- https://debian-handbook.info/browse/es-ES/stable/sect.apparmor.html
- https://wiki.archlinux.org/title/AppArmor
- https://wiki.archlinux.org/title/Security#Mandatory_access_control
- https://blog.cloudflare.com/de-de/linux-kernel-hardening/
- https://www.kernel.org/doc/html/latest/admin-guide/LSM/index.html
- https://documentation.suse.com/sles/12-SP5/html/SLES-all/cha-apparmor-intro.html

¿Preguntas?