Systemy operacyjne

Wykład 7: Tożsamość, autoryzacja i uwierzytelnianie

Pojęcia podstawowe

W systemach wieloużytkownikowych wymagana jest **kontrola dostępu** (ang. *access control*) do zasobów.

Procesy mają **tożsamość** (ang. *identity*) użytkownika, z reguły tego który je utworzył. W bloku kontrolnym procesu jądro trzyma niepodrabialne **kredencjały** (ang. credential).

Autoryzacja (ang. *autorization*) to proces sprawdzania, czy proces o danej tożsamości ma dostęp do zasobu.

Tożsamość należy potwierdzić przy pomocy mechanizmu **uwierzytelniania** (ang. *authentication*), np. podanie hasła, klucza prywatnego, odcisku palca.

Tożsamość procesu Uniksowego

- **użytkownik → `getuid(2)`**
- **grupa podstawowa → `getgid(2)`**
- **grupy rozszerzone → `getgroups(2)`**

Jądro przechowuje identyfikatory w postaci numerycznej!

W systemie istnieje wyróżniony użytkownik **`root`** (uid=0).

```
# id
uid=1000(cahir) gid=1000(cahir) grupy=1000(cahir),
24(cdrom),27(sudo),29(audio),44(video),46(plugdev),
108(netdev),123(vboxusers),999(docker)
```

Skąd się biorą nazwy użytkowników i grup?

Przypisanie nazw do identyfikatorów oddelegowane całkowicie do przestrzeni użytkownika. Obsługuje to baza danych usługi Name Service Switch.

Bazę danych użytkowników / grup można wylistować narzędziem `getent`!

Z reguły baza danych odnosi się do zawartości plików: **`/etc/passwd`**, **`/etc/groups`**, **`/etc/shadow`**.

Istnieje zestaw narzędzi systemowych do modyfikacji bazy danych np.: `useradd`.

Autoryzacja w systemach uniksowych

Proces może prosić jądro o przydzielenie zasobu z danej **przestrzeni nazw** → system plików, procesy, …

Jądro udziela lub odmawia dostępu (autoryzacja) do:

- zasobu plikowego na podstawie kredencjałów, trybu dostępu (O_RDWR), limitów systemowych, oraz oczywiście metadanych pliku.
- wysłania sygnału do innego procesu, wyłącznie na podstawie kredencjałów procesu źródłowego i docelowego

Pytania?