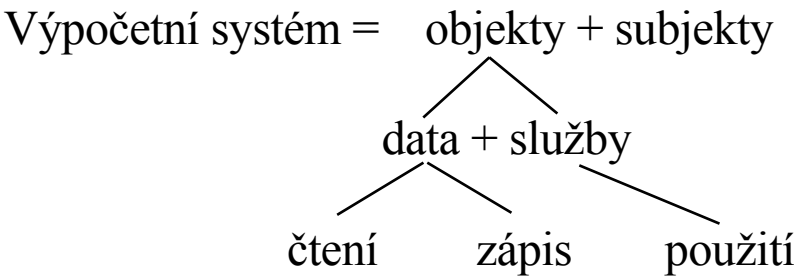
16. Ochrana a bezpečnost

Problém důvěryhodnosti výpočetního systému - zabránění zneužití:

- záměrné útoky
- nechtěné zásahy

Ochrana - metody, algoritmy zabránění zneužití
Bezpečnost - stav, ke kterému směřuje ochrana + další metody



Typy zneužití:

1. Prozrazení informace - ochrana objektů proti čtení nepovolanými subjekty, zajištění soukromí
2. Ztráta (neautorizovaná změna) informace - ochrana objektů pro modifikaci nepovolanými subjekty, ochrana systému
3. Neautorizované použití objektu - obejití účtování, sledování, omezení přístupu
4. Neposkytnutí služby - přetížení, porucha, havárie

Metoda dosažení bezpečnosti systému:

1. **Autentizace** - ověření identity subjektu
2. **Autorizace** - povolení/zabránění použití objektu danému subjektu
3. **Vynucení** - programové a administrativní postupy k dosažení

1. Autentizace

Externí - omezení přístupu k výpočetnímu systému
Interní - ověření identity uživatele (znalostí tajné informace)
Kombinované - identifikační karty, fyzické vlastnosti

Útoky na autentizaci:
obejití autentizace - vkládání paketů, zadní vrátka
překonání autentizace - odposlouchávání, monitorování, falešná autentizace, pokus/omyl, skryté komunikační kanály, odpadky

Použití hesel:
- nesmí být uloženy v dekódovatelném tvaru
- algoritmus ověření nesmí být rychlý (útoky hrubou silou)
- hesla nesmí být krátka a odhalitelná (útoky hrubou silou)

Unix: heslo+salt → klíč pro 25xdes(0) → signatura
Win LM hash: substr(upper(heslo), 1..7/7..14) →
2 x klíč pro des("KGS!@#$%")
Windows NT: md4(heslo) → signatura
BSD/Linux: md5(heslo) → signatura ($1)
Novější verze: blowfish ($2), SHA-256 ($5), SHA-512 ($6)

2. Autorizace

Co může kdo dělat

plný zápis = matice přístupových práv

	objekt			
subjekt	1	2	3	...
A	rw	r		
B		rw		
C			r	

Obtížná reprezentace, řešení:

1. **Přístupový seznam** (ACL, Access Control List) - součástí objektu jsou práva subjektů v seznamu (subjektů, právo)

Unix: majitel, skupina, ostatní
POSIX 1003.1e, SVR4, SOLARIS, NetWare: diskrétní uživatel, skupina (setfacl, getfacl)

2**. Deskriptory, lístky** (capability) - součástí subjektů jsou práva ke zpřístupněným objektům - ochrana interních struktur v jádře, deskriptory otevřených souborů, tabulka stránek, apod.

3. Vynucení

Implementace autentizace a autorizace

Problém imlementace:

- implementace vynucení by měla být co nejjednodušší, aby se dala verifikovat, což ale znamená primitivnější ochranu
- komfortní ochrana - složitá, problém verifikace
- skrytý komunikační kanál - nepřímé prozrazení chráněné informace

1. Prostředí procesu
- registry
- stav procesoru (chráněný režim činnosti procesoru)
- ochrana V/V

2. Primární paměť
- omezení přístupu do primární paměti
- nastavení omezení musí být privilegovaná instrukce

3. Sekundární paměť
- přístup přes jádro systému - využití autorizace v jádře
- mimo jádro - problém autentizace

Ochrana Unixu (man 2 intro)

Autentizace:
Databáze účtů */etc/passwd* – musí být čitelná všem (*ls*, *ps*)
Hesla – */etc/shadow*, */etc/master.passwd* – není přístupná
Realizace – *login* ověří jméno a heslo, nastaví UID a GID vzniklému procesu, další procesy uživatele toto nastavení dědí.

Autorizace:
Identita majitele uložena pro každý proces, při *fork()* se dědí, při *exec()* s výjimkou SUID, SGID programů také.
UID – správce *root* (uid 0), normální uživatel (uid <> 0)
GID – skupina
EUID – efektivní UID, pro SUID program počátečně majitel souboru programu
EGID – efektivní GID, pro SGID program počátečně majitel
Autorizace procesu je odvozena od **EUID, EGID.**

SUID program – patří majiteli PRIV_UID, spustí jej proces s nějakým UID, nastaví se RUID=UID, EUID=PRIV_UID
Problém: program potřebuje pracovat jako RUID:

```
... (práce jako PRIV_UID)
seteuid(RUID);              (RUID = getuid())
... (práce jako RUID)
```

Problém: jak se vrátit zpět k privilegované identitě?
POSIX 1003.1 – saved UID = počáteční EUID, saved GID dtto, lze při EUID=RUID provést:

```
seteuid(PRIV_UID);
```

Volání jádra – omezení na *setuid()*, *seteuid()*, *mount()*, *chroot()*, *mknod()*, *nice()*, *setrlimit()*, *settime()*, *ioctl()*, *mlock()*, ...

Systém souborů – majitelství a práva na úrovni i-uzlu:

- pouze majitel smí změnit práva a majitelství
- přístupová práva R,W,X pro majitele, skupinu a ostatní
- *chown*, *chgrp*, *chmod* – ochrana při předávání SUID
- ACL (POSIX 1003.2c) – *setfacl*/*getfacl*

MS Windows NT a novější

Autentizace:

Databáze účtů **SAM** (Security Accounts Manager), obsahuje login, hesla, RID (lokální UID, 32 bitů), SID (globální ID), plné jméno, domácí adresář, umístění profilu, atd.

Po autentizaci uživatele vznikne proces (*explorer*)+*access token*, ten se kopíruje při spouštění procesů, může být také modifikován privilegovanými procesy.

Autorizace:

Access token je předáván při volání služeb jádra, obsahuje SID, skupinové SID, práva ke službám (*privileges*), implicitního majitele a ACL (pro objekty vytvořené procesem). Každý zpřístupňovaný objekt má *Security Descriptor* obsahující majitele (SID), *System ACL* (audit) a *Discretionary ACL* (vlastní práva). Při přístupu se kontroluje zda proces může použít danou službu a zda je pro dané SID/skupinové SID povolena v ACL objektu.