Otázka 28

Zadání

Identita uživatelů, procesů a souborů v OS Unix, přístupová práva a jejich nastavení

Předmět: YD38UOS

Základní pojmy

uživatel – fyzická osoba užívající počítačový systém
uživatelské jméno – krátký jednoslovný (bez mezer) textový řetězec, kterým se uživatel identifikuje systému; uživatelské jméno rovněž určuje jméno domovského adresáře uživatele
přístupové heslo – řetězec znaků, který je známý pouze danému uživateli a který slouží k jeho autentizaci vůči počítačovému systému
domovský adresář – je adresář v systému, který patří danému uživateli; ten k němu má plný přístup (čtení, zápis, mazání, prohlížení) a má v něm umístěny svoje uživatelské soubory, jakožto i profilové konfigurační soubory; v unixových OS je jeho jméno shodné s uživatelským a obvykle se nachází v adresáři */home*

Identita uživatelů, procesů a souborů v OS Unix

Identita uživatele

Uživatel se identifikuje operačnímu systému svým uživatelským jménem a heslem (případně certifikátem, je-li v rámci systému nainstalována podpora infrastruktury PKI) – to se nazývá vnější identitou, protože je hmatatelná, zapamatovatelná a použitelná člověkem. V rámci procesů operačního systému jsou systémem samotným namísto jména užívána celá, tzv. identifikační čísla, která určují daného uživatele a jeho skupinu a slouží k autorizaci procesů na základě identity uvnitř systému (tedy oveření práv pro přístup k souborům a dovoleným operacím a spouštění jiných procesů). To je nazýváno vnitřní identitou, protože s ní pracuje pouze počítač skrytě před uživatelem.

Činnosti uživatele a systému při přihlášení:

- uživatel identifikuje systém, na který se chce přihlásit – buď se může přihlašovat k lokální konzoli (tzn. sedí fyzicky přímo u terminálu, tedy klávesnice a obrazovky, daného počítače) nebo se může přihlašovat vzdáleně prostřednictvím programu emulujícího terminál (za použití protokolů telnet nebo ssh); pokud se přihlašuje vzáleně, je třeba, aby znal jméno nebo IP adresu kýženého počítače.
- pokud uživatel pracuje na textovém terminálu, je třeba mít na obrazovce výzvu (prompt) ' *login:* ', což značí, že je terminál připojen; pokud tomu tak není, je třeba stisknout několikrát po sobě ENTER, případně CONTROL-q
- pokud se uživatel připojuje vdáleně, je třeba se připojit pomocí utility ***telnet*** nebo ***ssh***; protokol TELNET přenáší veškeré údaje mezi lokálním a vzdáleným počítačem, včetně uživatelského jména a hesla jako čistý otevřený text, proto již dnes není považován za bezpečný a nepoužívá se; SSH protokol oproti němu používá silné šifrování a je dnes silně doporučován – jinak oba protokoly pracují identicky
- jakmile má uživatel před sebou výzvu k zadání přihlašovacího jména *login:*, může ho zadat a poté je vyzván k zadání odpovídajícího hesla; to je možné zadat standardně 3X, poté systém uživatele od konzole odpojí a je třeba se připojit znovu
- systém ověří vložené informace oproti databázi uživatelských účtů a v případě úspěchu spustí přihlašovací shell a nastaví pro tento proces následující:
    - ➔ pracovní adresář na domovský adresář daného účtu
    - ➔ reálné číslo uživatele RUID = UID
    - ➔ efektivní číslo uživatele EUID = UID
    - ➔ reálné číslo primární skupiny RGID = GID
    - ➔ efektivní číslo primární skupiny EGID = GID

Pro úspěšné přihlášení musí být na daném systému vytvořen příslušný uživatelský účet!

Uživatelský účet

Uživatelské účty jsou vytvářeny správcem (administrátorem) systému. Účet jednoznačně definuje uživatele a jeho pracovní prostředí. Ve většině případů se účty zapisují do souboru */etc/passwd*. Ten slouží k uložení informací nutných pro přihlášení uživatele a k překladu UID na jméno a naopak. V tomto souboru byla původně uložena i hesla, ale kvůli bezpečnosti byla přemístěna do souboru */etc/shadow*. Tento soubor obsahuje hesla v zašifrované podobě – přesněji obsahuje jejich hashe, tedy jejich otisky vytvořené jednocestnou matematickou funkcí, takže z nich nelze zpětně získat původní heslo v textové podobě. Pro každý uživatelský účet je určena jedna řádka souboru, která sestává ze sedmi položek.

Struktura řádku (záznamu) souboru */etc/passwd* – jednotlivé položky jsou odděleny dvojtečkou, příklad:

```
manoupet:x:1001:1001:Petr Manoušek,+420604648571:/home/manoupet:/bin/bash
```

Každý záznam tedy obsahuje tyto položky:

uživatelské (také přihlašovací) jméno

Uživatelské jméno je textový řetězec jednoznačně identifikující konkrétního uživatele operačního systému a je většinou tvořeno skupinou 3 - 8 znaků. Uživatel se jím hlásí do systému. Jméno musí vždy začínat písmenem a zapisuje se malými písmeny. Může obsahovat i číslice. Jméno musí být pro každou osobu unikátní.

heslo

Tak jako i v jiných operačních systémech, kde se používá heslo, tak i v Unixu si heslo volí uživatel. Toto heslo se v počítači ukládá do souboru */etc/passwd* nebo */etc/shadow,* jak již bylo zmíněno. Heslo je do těchto souborů vkládáno v zakódované (hašované) podobě. Druhý soubor, tedy /etc/shadow je pro ukládání hesel vhodnější, jelikož u něj nic nebrání nastavit právo pouze pro čtení - a to ještě pouze pro správce systému. Nikdo jiný a nepovolaný se potom k heslům nedostane. Pokud je v souboru */etc/passwd* u hesla uveden znak x, znamená to, že heslo je uloženo v zakódované podobě v souboru */etc/shadow.* Hesla pro skupiny se ukládají do souboru */etc/gshadow.* Heslo účtu není povinné a uživatelé tedy nejsou povinni si svůj účet chránit heslem, ale záleží na konkrétním nastavení systému. Je dokonce i možné, aby systém hlídal minimální délku a komplexnost voleného hesla. Pokud má k počítači přístup více než jeden uživatel, je velice vhodné si heslo vytvořit. Toto heslo by pak mělo mít alespoň 6 znaků a nemělo by se skládat pouze z písmen, ale mělo by obsahovat i číslice a další znaky.

identifikační číslo uživatele (UID, ID uživatele, User IDentification)

Jedná se unikátní celé číslo, které je přiřazeno každému uživateli a kterým je tento identifikován uvnitř systému. Unix ho používá k identifikaci namísto uživatelského jména. Je to výhodnější, protože je rychlejší najít uživatele podle určitého UID a ten potom předávat různým procesům, než porovnávat textové řetezce, které tvoří kompletní jméno. Používá též se uvnitř systémových datových struktur. Při komunikaci s uživatelem se přes soubor */etc/passwd* překládá do textové podoby uživatelského jména. Číslo je obyvkle přidělováno administrátorem nebo automaticky systémem při vytváření účtu. UID=0 (nula) je vyhrazeno pro tzv. privilegovaný účet, tradičně se jménem root.

identifikační číslo primární skupiny (GID, ID skupiny, Group Identification)

Uživatel též spadá do nějaké uživatelské skupiny. Ta je zde zaznamenána jako unikátní celé číslo, kterým je skupina identifikována uvnitř systému. Každý uživatel sice může být ve více skupinách, ale primární je pouze ta, která je uvedena v souboru */etc/passwd*. Ostatní skupiny jsou zaznamenány v souboru */etc/group*. Členství v primární skupině již nemusí být zaznamenáno v souboru */etc/group*. Uživatel se může přepínat z jedné skupiny do druhé a to za použití příkazu *newgrp*. Všechny soubory vytvořené uživatelem jsou defaultně přístupné této skupině.

Soubor */etc/group potom* obsahuje seznam všech skupin existujících v systému – jeden řádek pro jednu skupinu. Jednotlivé položky na řádce jsou opět odděleny dvojtečkou. U každé skupiny jsou uvedeny tyto údaje: jméno skupiny:heslo:GID:eventuelně seznam členů, má-li skupina nějaké.

komentář

Popis uživatelského účtu – opět se jedná o textový řetězec; obvykle zahrnuje plné jméno uživatele a kontaktní údaje, včetně emailu, telefonního číslo, případně čísla kanceláře.

domovský adresář

Je to absolutní cesta k adresáři, která se po přihlášení nastaví jako běžný (aktuální) adresář a který je nadále označován jako domovský adresář uživatele. Je to adresář, ve kterém se interpret bude nacházet bezprostředně po přihlášení daného uživatele a je na něj nastavena proměnná HOME.

command / shell

Absolutní cesta k programu, který je spuštěn při každém přihlášení uživatele. Obvykle se jedná o výchozí shell, který vytváří textové uživatelské prostředí CLI a interpret – dnes obvykle *bash*, ale může se jednat i o zcela jiný program.

Struktura řádku (záznamu) souboru *`/etc/shadow`* – jednotlivé položky jsou odděleny dvojtečkou
`username:password:lastchg:min:max:warn:inactive:expire:flag`

Obsahuje:

- ➔ jmeno daneho uzivatele
- ➔ zakodovane heslo
- ➔ další parametry jako datum expirace, poslední změny...

Struktura řádku (záznamu) souboru *`/etc/group`*
`skupina:x:GID:seznam uzivatelu`

Obsahuje:

- ➔ jméno skupiny
- ➔ unikátní číslo skupiny (GUID)
- ➔ seznam uživatelů spadajících do skupiny

Soubor slouží k překladu GID na jméno skupiny a naopak a definici sekundárních skupin.

Identita procesů

Unixové procesy mají efektivní (EUID, EGID), reálné (RUID, RGID nebo častěji jen UID, GID) a uložené (SUID, SGID) ID. Za normálních okolností jsou identické, ale v *setgid* procesech jsou různé.
Jsou to celá čísla sloužící k autorizaci procesu uvnitř systému (ověření práv k provádění operací se soubory a práv ke spouštění jiných procesů).

Reálná identita procesu (RUID, RGID)
Systém si pamatuje, pod jakým uživatelským účtem se uživatel původně přihlásil (ať už lokálně nebo vzdáleně pomocí ssh,…)
Lze ji zjistit např. pomocí příkazů: *who am i , ps -o ruid,rgid,comm*

Efektivní (aktuální) identita procesu (EUID, EGID)
Slouží k autorizaci procesu uvnitř systému (např. při vyhodnocování přístupu k souborům, …)
Bezprostředně po přihlášení je reálná a efektivní identita totožná - efektivní identitu lze změnit např. pomocí příkazu *su* nebo speciálních práv binárních souborů.
Lze ji zobrazit např. pomocí následujících příkazů: id, ps –o uid,gid,comm

Změna identity procesu
Identitu procesu nastavuje jádro (kernel) při startu procesu nebo ji mění na žádost procesu. Obvykle jsou RUID a EUID, resp. RGID a EGID stejná a dědí se od rodičovského procesu. Ve zvláštních případech se nedědí, ale nastavují se všechna nebo jen některá ID:

- ➔ při přihlášení (pomocíprocesůlogin/dtlogin)
- ➔ pomocí příkazu su
- ➔ u binárních programů s nastaveným suid bitem se mění EUID
- ➔ u binárních programů s nastaveným sgid bitem se mění EGID

Příkaz su
Příkaz spustí nový shell pod novou identitou. Původní shell nekončí, po odhlášení se ze subshellu vyvolaného pomocí *su* je řízení vráceno původnímu nadřízenému shellu a jeho běh pokračuje. Je-li su volán běžným uživatelem, vyžaduje heslo, pokud rootem, tak ne. Pokud je za příkazem uveden přepínač -, provede přihlašovací skripty a nastaví prostředí uživatele root (jako jsou shellové proměnné, prohledávací cesty apod.). Vynecháme-li přihlašovací jméno, doplní se automaticky jméno *root.*
Jednoduše řečeno, příkazem se můžeme přepnout na jiného uživatele včetně roota (pokud známe jejich hesla) bez nutnosti odhlášení a opětovného přihlášení.

`syntax:` `su [ - ] [ uživatelské jméno ]`

Identita souborů

Soubor je datová oblast v paměti, na disku, či abstrakce jiného zařízení schopného zpracovávat datové proudy. Tvoří ho tedy posloupnost bytů. Ta je identifikována jménem, kterým lze na ni odkazovat. V Unixu je v podstatě každé bytově orientované zařízení považováno za soubor; z toho plyne množství typů souborů, které se v souborovém systému mohou vyskytovat.

Typy souborů

obyčejné soubory – všechny běžné datové soubory, v kterých jsou uložena data programů či uživatelů (textové a binární soubory)

adresáře - speciální soubory představující adresáře
symbolické linky - soubory, které obsahují odkaz na jiný soubor
pojmenované roury (named pipes) – poskytují mechanismus komunikace mezi procesy tím, že směrují výstup jednoho procesu na vstup jiného; to umožňuje mimo jiné řetězit příkazy a vytvářet tak nové funkce
sockety – další prostředek komunikace mezi procesy, kdy komunikující procesy nemusí být na stejném počítači; na rozdíl od rout jsou plně duplexní
soubory zařízení – představují zařízení připojená k systému tím, že reprezentují rozhraní mezi daným zařízením a operačním systémem

Jména souborů a adresářů

Maximální délka názvu souboru závisí na konkrétní implementaci systému souborů, obvykle bývá 255 znaků
Mezi nepovolené znaky patří v podstatě pouze lomítko /, protože slouží jako oddělovač. Všechny ostatní znaky je sice možné použít, nicméně jistou množinu znaků je důrazně doporučeno nepoužívat, protože slouží v shellu ke zvláštním účelům. Do ní patří speciální znaky, které mají v rámci shellu zvláštní význam a jsou jím odlišně interpretovány. Na takto pojmenované soubory je potom komplikované se běžným způsobem v příkazovém řádku odkazovat.

Patří mezi ně:

```
* ? [ ] ( ) \ “ ‘ ' ` + , ; & | $ < > { } ^ # % ! ~
```

Dále nebývá doporučováno používat bílé znaky, mezi něž patří mezera, tabelátor a znak nového řádku.

Pro názvy souborů se doporučuje používat pouze: alfanumerické znaky, tečky, pomlčku (ne na začátku), podtržítko.
Názvy souborů v unixových systémech jsou tzv. case-sensitive, tj. rozlišují se velká a malá písmena.
Jména začínající tečkou jsou skryté soubory/adresáře, které se standardně nezobrazují ve výpisu souborů.
Rezervovanými jmény jsou porom: tečka . (aktuální adresář) a dvě tečky .. (nadřazený adresář – jedna úroveň výše).

Každý soubor má svého vlastníka. Vlastníkem souboru je automaticky ten, kdo jej vytvořil. Vlastník může nastavovat přístupová práva pro všechny uživatele. Vlastníka lze změnit pomocí příkazu *chown.*

Každému souboru je též přiřazena jedna skupina. Defaultně se vytvářenému souboru přiřadí primární skupina jeho vlastníka. Lze ji měnit příkazem *chgroup.*

Root může měnit přístupová práva ke všem souborům a může rovněž měnit vlastníka a skupinu souboru. Identifikace vlastníka je dle UID, identifikace skupiny je dle GID.

Přístupová práva a jejich nastavení

Přístupová práva v Unixu umožňují ve víceuživatelském systému definovat přístup k adresářům a souborům na základě uživatelských účtů a skupin. Kontrola přístupu umožňuje na systémové úrovni zabránit uživatelům, aby záměrně nebo omylem data poškodili nebo zneužili.
Přístupová práva představují atributy každého souboru nebo složky. Ty určují, jaká práva nad tímto souborem/složkou určití uživatelé mají. Tyto atributy se zapisují přímo do tabulky i-nodů (viz. předchozí téma).

V i-uzlu každého souboru a adresáře jsou zaznamenány:

- typ souboru (soubor, adresář, zařízení, symlink...)
- počet odkazů na soubor (tj. na tentýž i-uzel)
- vlastník souboru (UID, owner)
- vlatnická skupina souboru (GID, group)
- velikost souboru v bytech
- datum a čas poslední změnu souboru
- jméno souboru
- přístupová práva k souboru

Tyto informace můžeme vypsat pomocí příkazu `ls -l`

```
# ls -l
total 3
drwxr-xr-x+ 1 manousek root  0 Nov  2 16:23 adresar1
drwxr-xr-x+ 1 manousek root  0 Nov  2 16:23 adresar2
-rwxr--r--  1 manousek root  7 Nov  2 16:23 soubor1.sh
-rw-rw-r--  1 manousek root 26 Nov  2 16:24 soubor2.txt
-rw-r--r--  1 manousek root 49 Nov  2 16:24 soubor3.odt
```

V ukázce vidíme, že v aktuálním adresáři se nacházejí dva adresáře a tři běžné soubory. Všechny patří uživateli manousek a spadají do skupiny root. Všechny soubory mohou čísti všichni uživatelé. Do prvního a třetího souboru může zapisovat pouze jeho vlastník, do druhého i všichni uživatelé spadající do skupiny root. První soubor lze jeho vlastníkem navíce spouštět.

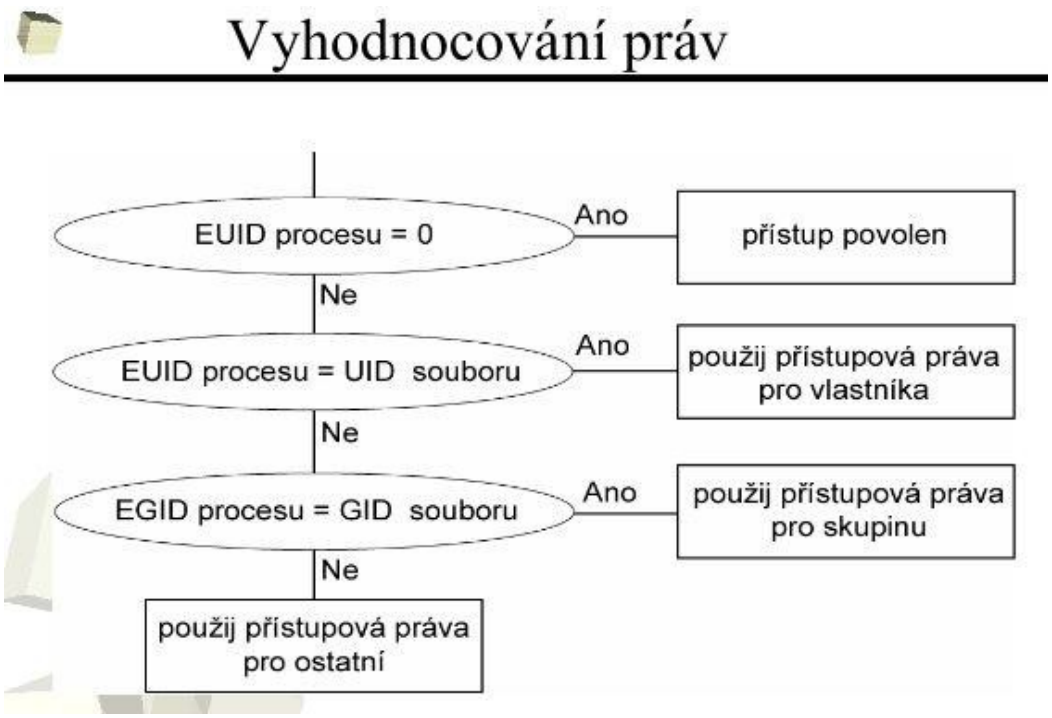
Na místě prvního znaku mohou být:

```
- = běžný soubor
d = adresář
b = blokově orientovaný speciální soubor
c = znakově orientovaný speciální soubor
l = symbolický odkaz
p = pojmenovaná roura
s = doménový socket
```

Přístupová práva lze nastavovat trojí: čtení (**r**ead), zápis (**w**rite), spouštění (e**x**ecute).
A lze je nastavovat pro tři typy uživatelů: pro vlastníka (**u**ser), skupinu (**g**roup), ostatní (**o**ther).

Následují tři trojice znaků **rwx** za sebou. Pokud má soubor dané právo nastaveno, vídíme odpovídající písmenko, pokud ne, vídíme jen pomlčku. Tři skupiny odpovídají třem typům uživatelů.

Následující schéma ukazuje, jak jsou efektivní práva vyhodnocována operačním systémem:



Zápis oprávnění

V unixových systémech se práva můžou zapisovat dvěma způsoby: buď oktalově, což představuje absolutní číselnou hodnotu odpovídající danému nastavení nebo pomocí symbolického zápisu, kdy se jednotlivé typy práv ke stávajícím buď přidávají nebo odebírají.

Symbolické i číselné oktalové hodnoty ukazuje následující tabulka:

Typ práva	Symbolické vyjádření	Oktalové vyjádření
Čtení	r (Read)	4
Zápis	w (Write)	2
Spuštění	x (eXecute)	1

U oktalového vyjádření je výsledná hodnota součtem všech dílčích práv.

Následující možné hodnoty potom tedy mají význam:
0 – žádné právo nepřiděleno
4 – právo číst (pouze, nic jiného)
2 – právo zapisovat (pouze, nelze nic jiného – ani číst)
1 – právo spouštět (pouze, soubor jinak nelze ani prohlížet)
6 – přiděleno právo číst a zapisovat (4+2)
5 – přiděleno právo číst a spouštět (4+1)
3 – přiděleno právo zapisovat a spouštět (2+1)
7 – všechna práva (4+2+1)

Význam oprávnění

Oprávnění pro soubory a adresáře se významově poněkud liší, jak zachycuje následující tabulka:

	Soubor	Adresář
Read	čtení ze souboru	čtení adresáře (výpis obsahu)
Write	zápis do souboru (změna obsahu, délky)	zápis do adresáře (vytváření, mazání a přejmenování souborů i podadresářů)
eXecute	spuštění (program, skript)	vstup do adresáře

Příklady zápisu oprávnění

700

Vlastník	Skupina	Ostatní
r w x	r w x	r w x
4 2 1	0 0 0	0 0 0

600

Vlastník	Skupina	Ostatní
r w x	r w x	r w x
4 2 0	0 0 0	0 0 0

755

Vlastník	Skupina	Ostatní
r w x	r w x	r w x
4 2 1	4 0 1	4 0 1

644

Vlastník	Skupina	Ostatní
r w x	r w x	r w x
4 2 0	4 0 0	4 0 0

700 - vlastník souboru jej může číst, zapisovat do něj a spouštět ho; u adresáře do něj může zapisovat a prohlížet ho; jinak nikdo jiný nemůže nic

600 - vlastník souboru ho může číst a zapisovat do něj; jinak nikdo nic

755 - tabulka právo čtení a otevření adresáře mají všichni, měnit data může pouze vlastník adresáře
- u souboru má vlastník všechna práva, všichni ostatní jej můžou spouštět, ale ne do něj zapisovat

644 - právo čtení souboru mají všichni, měnit data může jen vlastník souboru

Příkaz na změnu přístupových práv

Na změnu práv slouží príkaz *chmod*. Měnit práva souboru nebo adresáře může buď jejich vlastník anebo root.

```
chmod [-R] práva seznam_souborů
```

Parametr -R(recursive) znamená, že změna práv se aplikuje na všechny soubory a podadresáře daného adresáře.

Jak již bylo řečeno, měnit práva můžeme buď absolutně pomocí oktalového zápisu nebo pomocí symbolického zápisu:

oktalový zápis: `chmod 706 soubor`

Zde přímo zapisujeme výsledná práva v číselné oktalové podobě.

symbolický zápis: `chmod u+x,g-r,o+w soubor`

Zde přidáváme právo spouštění pro vlastníka, odebíráme právo zápisu skupině a povolujeme zápis všem ostatním.

Písmeno před znaménkem vždy znamená, kterému typu uživatelů daná práva nastavujeme.
(u - vlastníkovi, g – skupině, o – všem ostatním, a – všem bez výjimky, tedy všem třem typům)
Znaménko za písmenem značí, zda jsou oprávnění přidávána (+), odebírána (-), či absolutně nastavena (=).
Písmenko za znaménkem určuje, která práva jsou pro příslušný typ uživatelů nastavována.
(r – čtení, w – zápis, x – spouštění či u adresářů listování)

Změna vlastnictví souboru

Může měnit pouze root
Vlastnictví (i skupinové) lze měnit příkazem chown, skupinové příkazem chgrp.

chown [-R] vlastník [:skupina] seznam_souborů

chgrp [-R] skupina seznam_souborů

Maska přístupových práv

Jedná se o globální nastavení systému, přesněji prostředí daného uživatele. Definuje přístupová práva nově zakládaných souborů/adresářů. Hodnota masky je součástí procesu (podobně jako EUID,EGID,…) a je dědičná. Lze ji vypsat a měnit příkazem umask. Přístupová práva vzniknou množinovým rozdílem výchozí hodnoty a masky. Výchozí hodnota je 666 pro soubory a 777 pro adresáře.

maska	soubor	adresář	poznámka
000	666	777	odpovídá výchozí hodnotě, Nebezpečné
022	644	755	obvyklé nastavení
027	640	750	vyšší bezpečnost
077	600	700	největší restrikce
066	600	711	kompromisní řešení

Platí vzorec: **maska – soubor/adresář = výchozí hodnota**

Speciální oprávnění

Speciální oprávnění mění standardní chování systému, což je výhodné v některých speciálních případech.

SUID (setuid)

Za standardních okolností dědí potomek (nový proces) oprávnění svého rodiče. Někdy je však nutné, aby měl spuštěný program jiná (vyšší) oprávnění. Pokud je na souboru s programem nastaven SUID bit, neběží spuštěný program s právy rodiče, ale s právy vlastníka tohoto souboru. Používá se v případech, kdy chceme uživateli umožnit provedení akce, na které by potřeboval jiná nebo vyšší oprávnění.

Příklad - změna hesla

Hesla uživatelů jsou uložena v souboru /etc/passwd nebo /etc/shadow, do kterých běžný uživatel nemůže zapisovat (soubor shadow nemůže dokonce ani číst). Při změně hesla je ale potřeba změněné heslo do těchto souborů zapsat. Proto má program /usr/bin/passwd nastaven SUID bit a patří uživateli root. Po spuštění běží program passwd s právy roota a heslo může být do příslušného souboru zapsáno.

Změna uživatele

Program běžného uživatele nemůže změnit svoje oprávnění. Může to však udělat program běžící s právy roota. Proto má program su nastavený SUID bit a patří uživateli root'. Po zadání správného hesla je spuštěn nový shell, který je nastaven na nová oprávnění.

SUID bit neznamená, že program poběží s právy uživatele root. Patří-li program jinému uživateli, bude po spuštění běžet s právy tohoto uživatele. Nejběžnější je však tento způsob při poskytování administrátorských oprávnění. Proto musí být každý program se SUID bitem naprogramován s maximální obezřetností, aby neumožnil provést nějakou neoprávněnou činnost.

SGID (setgid)

Program s nastaveným SGID bitem se chová po spuštění podobně, jako u SUID bitu. Nepřebírá ale oprávnění majitele souboru, nýbrž oprávnění skupiny, které daný soubor s programem na disku patří. Při aplikaci SGID bitu na adresář patří všechny nově vytvořené soubory a adresáře do skupiny, která je shodná s nadřízeným adresářem (který má nastaven zmíněný SGID bit). Bez nastaveného SGID bitu patří nově vytvořené adresáře a soubory primární skupině uživatele (viz výše).

Příklad - skóre ve hře

Některé hry zapisují dosažené skóre do souboru, aby mohli hráči své výkony porovnat. Takový soubor by musel mít právo zápisy pro všechny uživatele v systému. Hráči by pak snadno mohli tento soubor měnit a své dosažené skóre neférově zvyšovat. Proto je program s hrou svěřen speciální skupině (např. games) a je mu nastaven SGID bit. Soubor se skóre pak bude mít právo zápisu přidělené jen skupině games. Do souboru se skóre tak spuštěná hra může zapisovat, kdežto uživatelé nemohou soubor měnit.

Jiný příklad - skupinový projekt

Uživatelé, kteří pracují na společném projektu patří do společné skupiny project a obvykle nemají tuto skupinu nastavenou jako primární (nebo si ji zapomenou před každou prací na projektu pomocí příkazu newgrp změnit). Vytvoří-li ve společném adresáři, kam mají na základě členství ve skupině project přístup, nový soubor nebo adresář, bude patřit jiné skupině. Ani při nastavení umask na hodnotu zajišťující skupině zápis tak nebudou kolegové moci soubory upravovat nebo soubory v nových adresářích mazat a přejmenovávat. Přidělit oprávnění všem není nikdy vhodné. Proto je na kořenový adresář projektu, který patří skupině project nastaven SGID bit. Nové soubory a adresáře tak automaticky patří skupině project a nové podadresáře mají nastaven SGID bit.

Sticky bit

Pokud má uživatel do adresáře právo zápisu, může v tomto adresáři i mazat. Nastavíme-li v adresáři sticky bit, bude uživatel smět smazat jen své vlastní soubory nebo adresáře. Ve starších verzích unixových systémů tento příznak na souborech sloužil k tomu, aby kód ukončeného programu zůstal ve swapu. Tímto způsobem bylo dosaženo rychlejšího startu často používaných programů. V dnešních systémech není sticky bit na souborech již delší dobu podporován, protože vlivem používání virtuální paměti a stránkování ztratil pro soubory význam.

Příklad - Adresář /tmp

Do adresáře /tmp mají všichni uživatelé v systému právo zápisu a odkládají si tam dočasné soubory (zejména to dělají různé programy). Pokud mají právo zápisu, mohou v adresáři i přejmenovávat a mazat soubory i adresáře, což by vedlo k tomu, že by si uživatelé mohli navzájem škodit. Nastavíme-li na tento adresář sticky bit a uživatelé budou vytvářet své soubory a adresáře s adekvátními oprávněními, bude problém vyřešen.

Zdroje

Slajdy předmětu Y36UOS
http://blog.milde.cz/post/88-pristupova-prava-v-unixu/ [http://blog.milde.cz/post/88-pristupova-prava-v-unixu/]
http://vse.ugic.net/doc/4IT425_Prednaska_I-3.pdf [http://vse.ugic.net/doc/4IT425_Prednaska_I-3.pdf]
Mistrovství v Linuxu
Wikipedia.org

Poslední úprava: 2011-10-30 22:48 autor: manoupet