VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

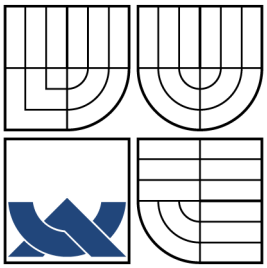
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNATIONS

ROZŠÍŘENÝ BEZPEČNOSTNÍ MODEL OPERAČNÍHO SYSTÉMU, ŘÍZENÍ PŘÍSTUPU

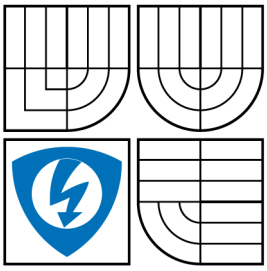
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE FILIP NOVOTNÝ
AUTHOR

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNATIONS

ROZŠÍŘENÝ BEZPEČNOSTNÍ MODEL OPERAČNÍHO SYSTÉMU, ŘÍZENÍ PŘÍSTUPU

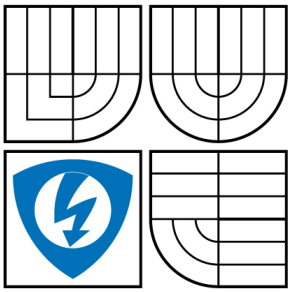
MANDATORY ACCESS CONTROL POLICIES OF OPERATING SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — FILIP NOVOTNÝ
AUTHOR

VEDOUCÍ PRÁCE — ING . TOMÁŠ PELKA
SUPERVISOR

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Novotný Filip	***ID:***	83415
Ročník:	3	***Akademický rok:***	2007/2008

NÁZEV TÉMATU:

Rozšířený bezpečnostní model operačního systému, řízení přístupu

POKYNY PRO VYPRACOVÁNÍ:

V bakalářské práci se student nejdříve seznámí se základním bezpečnostním modelem operacního sytému a utvoří ucelený přehled o soucasných rozšířených bezpecnostních modelech. Konkrétně se bude student zabývat metodami řízení přístupu MAC (Mandatory Access Control) a DAC (Discretionary Access Control).
Dále popíše postup tvorby přístupových pravidel v rámci SElinux, který je implementací řízení přístupu MAC. V závěru student shrne jednotlivé metody přístupu a uvede význam a uplatnění SELinux v linuxových systémech.

DOPORUČENÁ LITERATURA:

[1] MCCLURE, Stuart, SCAMBRAY, Joel, KURTZ, George. Hacking bez tajemství. 3. preprac. vyd. Brno: Computer Press, 2004. 632 s. ISBN 80-722-6948-8.

[2] MCCARTY, Bill. SELinux : NSA's Open Source Security Enhanced Linux. 1st edition. [s.l.] : O'Reilly, 2004. 254 s. ISBN 978-0596007164.

[3] BENANTAR, Messaoud. Access Control Systems : Security, Identity Management and Trust Models. 1st edition. [s.l.] : Springer, 2005. 261 s. ISBN 978-0387004457.

Termín zadání:	11.2.2008	***Termín odevzdání:***	4.6.2008

Vedoucí práce: Ing. Tomáš Pelka

prof. Ing. Kamil Vrba, CSc.
předseda oborové rady

UPOZORNĚNÍ:

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Filip Novotný
Bytem: Trávník 330/23, 75005, Přerov - Přerov I-Město
Narozen/a (datum a místo): 3.3.1985, Přerov

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 244/53, 60200 Brno 2
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   - ☐ disertační práce
   - ☐ diplomová práce
   - ☑ bakalářská práce

   jiná práce, jejíž druh je specifikován jako .....

(dále jen VŠKP nebo dílo)

Název VŠKP: Rozšířený bezpečnostní model operačního systému, řízení přístupu
Vedoucí/školitel VŠKP: Ing. Tomáš Pelka
Ústav: Ústav telekomunikací
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

- ☑ tištěné formě - počet exemplářů 1
- ☑ elektronické formě - počet exemplářů 1

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
   - ☑ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

....

Nabyvatel Autor

ABSTRAKT

Práce se věnuje ochraně dat pomocí řízeného přístupu. Srovnává rozdíl řízeného přístupu pomocí režimů DAC a MAC. Popisuje vývoj režimu MAC na základě aktuálních požadavků uživatelů. Řeší problematiku SELinuxu, který nabízí systému větší bezpečnost a je založen na povinném řízeném přístupu MAC. Je zde uvedena problematika tvoření politiky SELinuxu. Vlastní tvoření politiky je věnováno SELinux editoru, který je založen na zjednodušených pravidlech a přibližuje tím SELinux i běžnému uživateli.

KLÍČOVÁ SLOVA

MAC, DAC, RBAC, subjekt, objekt, identita, bezpečnostní kontext, doména, typ, role

ABSTRACT

The subject of this paper is data protection by means access control. It compares the difference between DAC and MAC access control and describes the progress of MAC due to user requierements. Next part is dedicated to SELinux, what offers more system safeness. SELinux is based on mandatory access control. Making own policy is dedicated to SELinux editor, which is composed of Simplified Policy and is easy to use for ordinary user.

KEYWORDS

MAC, DAC, RBAC, subject, object, identity, security context, domain, type, role

NOVOTNÝ F. *Rozšířený bezpečnostní model operačního systému, řízení přístupu*. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací, 2008. 52 s. Bakalářská práce. Vedoucí práce byl Ing.Tomáš Pelka.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Rozšířený bezpečnostní model operačního systému, řízení přístupu“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Tomášovi Pelkovi za pomoc a cenné rady při zpracování bakalářské práce.

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Řízení přístupu, neboli autorizace v přeneseném smyslu, má kořeny již v dávné minulosti. Stráže, brány a zámky jsou používány již od časů, kdy bylo třeba zamezit přístupu nežádoucím osobám k cenným věcem. Potřeba řízeného přístupu však pobízela k vynalezení toho, co považujeme za první světový bezpečnostní operační systém. V roce 1879, James Ritty vynalezl „nepodplatitelného pokladníka“, který se později stal známým jako registrační pokladna. Rittyho vynález omezil častý problém kradoucích zaměstnanců tím, že zaměstnanec měl přístup do pokladny jen tehdy, když zadal prodejní částku, kterou viděl i zákazník. Záznam o množství prodeje a celkové částky tak umožnil majiteli obchodu kontrolu, zda peníze nacházející se v pokladně, opravdu korespondují s celkovou vyúčtovanou částkou během dne.

V dnešní době je řízení přístupu pravděpodobně nejvíce nezbytným a nejvíce prostupujícím bezpečnostním mechanismem. Řízení přístupu se virtuálně objevilo ve všech systémech a vynutilo si velké architektonické a administrativní výzvy ve všech úrovních počítačových odvětví. Z obchodní perspektivy má řízený přistup potenciál hájit zájmy k optimálnímu sdílení a výměny zdrojů, ale stejně tak má odradit od zpronevěry nebo korupce hodnotných informací.

Řízení přístupu má mnoho forem. Slouží jednak k určení, zda uživatel má právo používat daný zdroj, ale také pomocí něj lze omezovat kdy a za jakých podmínek je tento zdroj používán. Například uživatel vlastní povolení k přístupu do sítě jen během pracovní doby. Některé organizace mohou stanovit více komplexní řízení, kdy například dva zaměstnanci provádějí vysoce riskantní operaci, jako například vypouštění řízené střely. Definice a modelovaní řízeného přístupu pramení z klíčových dokumentů z ranných sedmdesátých let a z vyvinutí RBAC (*Role-based access control*), který započal v devadesátých letech a pokračuje dodnes.

1 ŘÍZENÝ PŘÍSTUP

1.1 Účel a podstata řízeného přístupu

Kontrola přístupu je jen jedno z hledisek úplného řešení otázky počítačové bezpečnosti, ale zato nejvíce důležité. Pokaždé, když se uživatel přihlásí do víceuživatelského počítačového systému, je vynucena kontrola přístupu. K lepšímu pochopení podstaty přístupu je účelné zhodnotit rizika informačních systémů. Informační bezpečnostní rizika mohou být obecně rozčleněny do následujících třech typů: utajení, integrita a dostupnost. Ve zkratce je můžeme nazvat CIA (*confidentiality, integrity, availability*).

- *Utajení:* je to požadavek držet data v bezpečí a soukromí. Tato kategorie může obsahovat státní tajemství, důvěrné záznamy, finanční informace a bezpečnostní informace jako je například přístupové heslo.
- *Integrita:* je to požadavek na ochranu dat, aby nemohla být zaměněna nebo jakkoliv změněna neautorizovanými uživateli. Například mnoho uživatelů se chce ujistit, zda čísla bankovních účtů užívaná ve finančním softwaru nemohou být někým změněna, a že jen uživatel nebo autorizovaný bezpečnostní administrátor může tato hesla měnit.
- *Dostupnost:* je představa, že pokud je potřeba, tak informace jsou vždy dostupné.

Řízený přístup je rozhodující pro ochranu důvěrnosti a integrity dat. Podmínka pro důvěrnost požaduje, aby jen autorizovaná osoba mohla číst data a podmínka integrity znamená, že jen autorizovaná osoba má dovoleno data měnit. Řízený přístup neklade důraz na dostupnost dat, ale má jasnou důležitou roli: Útočník, který dosáhne neautorizovaného přístupu do systému, bude mít pravděpodobně problém tento systém zničit.

1.2 Autorizace a autentizace

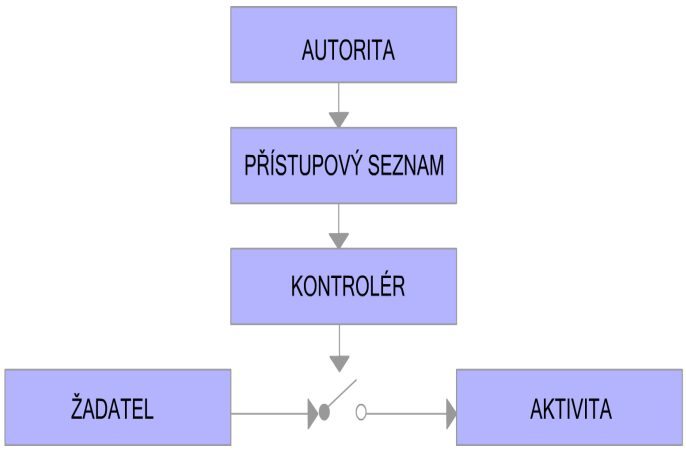
Autorizace a autentizace jsou základem pro řízený přístup. Mají odlišný význam, ale často bývají špatně vyloženy. Část toho pramení z blízkého vztahu mezi nimi; správná autorizace vlastně závisí na autentizaci.

Autentizace je proces, určující, že uživatelem požadovaná identita je skutečná. Každý počítačový uživatel je seznámený s hesly, jakožto nejznámějším druhem autentizace. Méně známé druhy autentizace zahrnují biometrii (např. otisk prstu) a různé čipové karty. Autentizace je založena na jednom či více z následujících faktorů:

- Tajná informace, znalost,
- vlastnictví předmětu,
- na základě osoby.

Většinou autentizace používá dva faktory, jako například bankomat, kdy požaduje bankomatní kartu a PIN zároveň. Je to mnohem efektnější způsob, protože při ztrátě karty by mohlo dojít ke zcizení hotovosti z účtu.

Zatímco autentizace je proces závisející na tom kdo jsme, autorizace je proces, který určuje co můžeme vykonávat. Autorizace odkazuje na „Ano“ a „Ne“, to znamená, jestli uživatel má povolený přístup do systému nebo nemá. Informační systém musí podporovat určitý vztah mezi uživatelovou identifikací a systémovými zdroji, například seznamem autorizovných uživatelů k daným zdrojům nebo zálohováním seznamu dostupných zdrojů ke každému uživateli. Mějme také na vědomí, že autorizace nutně závisí na správné autentizaci. Jestliže uživatelova identita není správná, pak zde není žadný způsob, jakým by mohl být povolený přístup do systému.



Obr. 1.1: Obecné schéma autentizace

1.3 Uživatelé, subjekty, objekty, operace a povolení

Téměř každý řízený přístupový model může být formálně uveden pojmy *uživatel, subjekt, objekt, operace, povolení* a jednotlivými vztahy mezi těmito entitami. Je důležité znát tyto pojmy, protože se s nimi v řízeném přístupu často setkáváme.

Pojem uživatel se týká lidí, kteří jsou začleněni do počítačového systému. V mnoha návrzích je možné pro jednoho uživatele, aby měl několik identit (ID) a tyto identity byly zároveň aktivní.
Situace, kdy uživatel komunikuje s počítačem, se nazývá *relace*.
Počítačový proces vyvolaný uživatelem se nazývá *subjekt*.

Každá uživatelská akce v počítačovém systému je představována nějakým běžícím programem. Uživatel může mít několik subjektů v operaci (na bežných multitaskingových operčních systémech), i když uživatel má jen jeden přístup (login) a jednu relaci. Například e-mailový klient může operovat na pozadí, stahující periodicky e-maily ze serveru, během toho co uživatel obsluhuje webový prohlížeč. Každý z uživatelských programů je subjekt a každý programový přístup je kontrolován, zda uživatel, který program vyvolal, má k tomu povolení. *Objekt* může být jakýkoliv zdroj dostupný v počítačovém systému včetně souborů, periférií (např. tiskárny), databází a i detailů jako jsou například individuální záznamy v databázi. Objekty jsou považovány za pasivní entity, které obsahují nebo přijímají informace.

Operace je aktivní proces vyvolaný subjektem. V začátcích modelů řízených přístupů, které byly spojovány přísně s informačním tokem (např. přístup pro čtení-zápis), se termín subjekt aplikuje ke všem aktivním procesům, ale RBAC (*Role-Based Access Control*) modely požadují rozdíl mezi subjektem a operací. Například uživatel bankomatu zadá správný PIN, kontrolní program operující na uživatelském rozhraní je subjekt, který může zahájit více než jednu operaci - zůstatek, výběr, vklad a další.

Povolení (privilegia) jsou autorizace vyvolávající určitou akci v systému. Tento pojem se vztahuje k nějaké kombinaci objektu a operace. Určitá operace použitá na dvou objektech reprezentuje dvě různá povolení, stejně tak dvě různé operace aplikované na jeden objekt reprezentuje dvě různá povolení.

2 REŽIMY PŘÍSTUPU DAC A MAC

2.1 Modely přístupových režimů DAC a MAC

V roce 1983 Ministerstvo obrany Spojených států zveřejnilo takzvanou „oranžovou knihu“ (*Trusted Computer System Evaluation Criteria - TCSEC*) [1]. Tyto standardy definovaly dva důležité režimy řízeného přístupu pro vojenské systémy: DAC (*discretionary access control*) a MAC (*mandatory access control*) . DAC je režim, ve kterém tvůrci nebo vlastníci souborů stanovují přístupová práva a subjekt s volným přístupem k informaci může tuto informaci šířit na další subjekt.

Jako takový je DAC pro armádu nedostatečným zabezpečením tajných dokumentů. K opravdovému zajištění bezpečnosti je potřeba MAC.

MAC kontrola zabezpečuje víceúrovňové zabezpečení. Klíčovým rysem MAC je to, že je požadován pro zprostředkování všech přístupů k objektům v systému. Protože tento systém řízeného přístupu zprostředkovává všechen přístup k objektům podle externě daných pravidel, uživatelé nemohou dále rozdávat povolení pro přístup k objektům. Díky tomu, že uživatelé jsou omezeni v určitých akcích, si můžeme být jisti, že systém setrvá v bezpečí bez ohledu na činnosti uživatele.

2.2 Srovnání modelů přístupů RBAC s DAC a MAC

Základním principem RBAC je schopnost specifikovat a uplatnit přístupová práva, a také usměrnit typicky obtížný proces správy autorizace. RBAC reprezentuje významné zlepšení v přizpůsobení a detailnější kontrole než standardy DAC a MAC.

DAC je definován jako kontrola přístupového práva a jako mechanismus, který umožňuje systémovým uživatelům povolit nebo zakázat ostatním uživatelům přístup k objektům, které jsou pod jejich kontrolou bez zakročení systémového administrátora.

V mnoha firmách, v průmyslu a civilní správě zaniká uživateli, který má k dané informaci přístup, právo ji vlastnit. Pro tyto organizace je vlastníkem systémových objektů společnost nebo agentura a nebylo by vhodné, aby uživatelé mohli měnit jejich přístupová práva. Za pomoci přístupu RBAC jsou přístupová práva založena na individuálních rolích uživatelů jakožto součástí organizace. To obsahuje specifikaci jejich povinností, zodpovědnosti a kvalifikace. RBAC právo je založeno na funkci nebo činnosti, kterou má uživatel povolenou v rámci organizace provádět. Uživatel nemůže předat jeho povolení na dalšího uživatele jen dle svého uvážení.

RBAC je někdy popisován jako forma MAC ve smyslu, že uživatelé jsou nevyhnutelně omezeni a nemají žádný vliv nad prosazováním ochranných práv v dané organizaci. Přístupy RBAC a MAC se však liší. MAC je definován jako omezený přístup k objektům, založený na citlivosti informace obsažené v objektu a na formální autorizaci objektů k takto citlivé informaci.

Jak je uvedeno v dokumentu TCSEC, MAC podporuje požadavky ministerstva obrany, vztahující se k neautorizovanému přístupu k tajným informacím a také pravidla k ochraně diskrétnosti citlivých dokumentů. Systémy, které podporují MAC strategii se dělí na vysoko úrovňové nebo nízko úrovňové.

2.3 Řízení přístupu metodou DAC

DAC [1] je prostředek, který zamezuje přístup k objektům založených na identitě uživatelů, případně skupin do kterých patří, nebo na obojím. Hlavní myšlenka tohoto přístupu spočívá v tom, že každý uživatel má plnou kontrolu nad všemi svými procesy a soubory. Některá práva k těmto procesům a souborům pak může poskytnout také jiným uživatelům. Slabým místem této filozofie je tzv. superuživatel. Jedná se o uživatele, který má administrátorská práva k celému systému. To v praxi znamená, že superuživateli se vůbec nekontrolují práva. Jestliže se tedy někomu podaří „ovládnout“ proces, který patří superuživateli, stává se neomezeným vládcem systému.

Mechanismus přístupu DAC směřuje k flexibilitě a je široce používán v komerčních a vládních sektorech. Během poloviny osmdesátých a devadesátých let mnoho organizací považovalo metodu řízeného přístupu DAC za standardní.

Ačkoliv jsou DAC mechanismy široce komerčně využívány, jsou neodmyslitelně považovány za slabé ze dvou důvodů. První důvod je, že udělovaní přístupu pro čtení je přechodné. Například když Petr udělí Pavlovi právo pro čtení souboru, tak nic nezamezí Pavlovi tomu, aby si zkopíroval obsah Petrova souboru do objektu, který spravuje Pavel. Pavel tak má možnost zpřístupnit tuto kopii jakémukoli dalšímu uživateli, aniž by o tom Petr věděl. Druhý důvod je, že mechanismus DAC nedokáže čelit útoku „trojského koně“. Protože programy dědí identitu od uživatele, který jej vyvolal. Pavel například může napsat pro Petra program, který se bude tvářit, že představuje užitečné funkce, ale za jeho běhu bude číst obsah Petrových souborů a zároveň ho kopírovat na místo, ke kterému mají oba dva přístup. Pavel pak může obsah těchto souborů přesunout někam, kde Petr k nim nebude mít přístup. Pavlův trojský kůň mohl dokonce obsah Petrových souborů zničit. Kdyby Petr chtěl zjistit co se stalo s jeho ztracenými daty, zjistil by, že si je dokonce smazal sám.

2.3.1 Matice pro řízený přístup

Z teoretického hlediska je přístupová matice tradičně užívána k zobrazení bezpečné situace přístupu. Je to vlastně pole obsahující řádek pro subjekt v systému a sloupec pro objekt. Níže uvedená tabulka 2.1 nám reprezentuje jednoduchou kontrolní přístupovou matici.

Tab. 2.1: Příklad přístupové kontrolní matice

subjekt/objekt	**Soubor 1**	**Soubor 2**	**Soubor 3**	**Proces 1**
Petr	Read,write	-	Write	-
Jana	-	Execute	-	Suspend
Pavel	-	Read	Read	-
Adam	Read	-	-	-

Zápisy v této matici specifikují operace nebo typ přístupu, který každý subjekt má k danému objektu. Základní funkcí každého systému přístupové kontroly je ujistit se, že mohou být provedeny jen ty operace, které jsou uvedeny v matici.

Z teoretického hlediska je přístupová kontrolní matice zajímavým řešením, pro systém s velkým počtem uživatelů a objektů bude velmi rozsáhlá a řídce zaplněna. Jako takový je systém přístupové kontroly málokdy realizovaný jako matice, ale skoro vždy je tak znázorňován. V dnešní době jsou zde dvě hlavní zastoupení přístupové kontrolní matice: seznam povolených přístupů (ACLs; access control lists) a seznam povolených operací (capability lists).

2.3.2 Seznam povolených přístupů a seznam povolených operací

V systému povolených operací je přístup k objektu povolen, jestliže subjekt, který požaduje přístup, je k danému objektu způsobilý. Způsobilost je chráněný identifikátor, který určuje jak objekt, tak přístupová práva, která jsou uživateli umožněna. V tomto systému se zápis do matice provádí po řádcích. Uvedená tabulka 2.2 reprezentuje seznam povolených operací. Každý subjekt je přiřazen k seznamu povolených operací, který obsahuje schválené operace ke všem zahrnutým objektům. Subjekt disponující povolením, je důkazem, že vlastní přístupová privilegia. Základní výhodou je, že je lehké zkontrolovat všechny přístupy, které jsou autorizovány pro daný subjekt.

Na druhou stranu je těžké zkoumat subjekty, které mohou přistupovat k jednotlivým objektům. To by znamenalo, vyzkoušet je všechny v každém seznamu povo-

Tab. 2.2: Seznam povolených operací

subjekt		
Petr	Soubor 1: Read, Write	Soubor 3: Write
Jana	Soubor 2: Execute	Proces 1: Suspend
Pavel	Soubor 2: Read	Soubor 3: Read
Adam	Soubor 1: Read	

lených operací. Z tohoto důvodu je těžké zrušit přístup k objektu. Proto je seznam povolených akcí kritizován ve spojení s metodou přístupu DAC, tudíž i komerčně nepopulární.

Seznam povolených přístupů (ACLs) sestavuje matici řízeného přístupu pomocí sloupců, které tvoří seznam uživatelů vztahujících se k chráněnému objektu. Každý objekt je spjat s ACL, který obsahuje subjekty a jeho schválené operace pro objekt. Seznam je kontrolován přístupovým kontrolním systémem, aby bylo určeno, zda je přístup povolen nebo zamítnut. Uvedená tabulka 2.3 se shoduje s výše uvedenou tabulkou 2.3.

Tab. 2.3: Seznam povolených přístupů

objekt		
Soubor 1	Petr: Read, Write	Adam: Read
Soubor 2	Jana: Execute	Pavel: Read
Soubor 3	Petr: Write	Pavel: Read
Proces 1	Jana: Suspend	

2.3.3 Atributy

Mechanismus atributů je podobný jako u seznamu povolených přístupů. Rozdíl je v tom, že namísto spojování uživatelů s operacemi, jsou atributy spojovány s objekty. Atributy dělí uživatelé do třech skupin:

- Vlastní: vlastník souboru,
- skupina: několik uživatelů sdílí společný přístup k souboru,
- ostatní: každý jiný uživatel kromě vlastníka nebo člena skupiny.

Přístupový kontrolní systém reguluje přístup k souboru pomocí atributů: čtení (r), zápis (w) nebo spuštění operace (x). Například složka může mít tyto atributy: (r w x) (r - x) (- - x)

Tento řetězec určuje, že vlastník má právo čtení, zápisu a spuštění k tomuto souboru; členové skupiny, kteří jsou spjati s tímto objektem mají právo čtení a spuštění tohoto souboru; všichni další uživatelé mají možnost právo spuštění. Znak pomlčky „-“ určuje, že odpovídající operační indikátor není přítomen, čímž efektivně zakazuje danou operaci se souborem pro určitou kategorii uživatelů.

Uživatel, který soubor vytvořil, se standardně stává vlastníkem souboru. Vlastník souboru je obvykle jediný kromě superuživatele, který může atributy souboru měnit.

Pouze jedna skupina má přístup ke každému souboru. Členství ve skupině kontroluje systémový administrátor.

Problém s těmito atributy je v mechanismu kontroly přístupu, který zcela nekoresponduje s maticí řízeného přístupu, proto systém nemůže přesně poskytnout přístup k objektu na jednotlivých platformách. Z těchto důvodů novější verze UNIXu a UNIXu jako operačních systémů obsahují ACL, tedy seznam povolených přístupů.

2.4 Řízení přístupu metodou MAC

Dokument TCSEC [1] definuje zásady přístupu MAC tak, že předcházejí problémům s trojskými koni. S ohledem na tuto zásadu jsou bezpečnostní úrovně přiřazeny uživatelům se subjekty působících ve prospěch uživatelů a objektů. Bezpečnostní úrovně mají hierarchickou a ne-hierarchickou součást. Hierarchická část určuje citlivost bezpečnostní úrovně. Například hierarchická součást může obsahovat:

- Nezařazené (U);unclassified,
- důvěrné (C);confidential,
- tajné (S);secret,
- vysoce tajné (TS);top secret.

Zatímco ne-hierarchická součást označuje typ bezpečnostní úrovně a může obsahovat NATO a NUCLEAR. Bezpečnostní úrovně jsou částečně řazeny podle nejvlivnějšího vztahu, často zapisovány pomocí znaku „$\geq$“. Například $TS \geq S \geq C \geq U$ a $S(NATO, NUCLEAR) \geq S(NUCLEAR) \geq S$. Úroveň bezpečnosti uživatele často vztahovaná k uživatelově autorizační úrovni, vyjadřuje úroveň důvěry udělené uživateli a musí vždy převládat nad bezpečnostními úrovněmi, které jsou přiřazeny uživatelově subjektu. Například Pavel, který je přiřazen k S (NUCLEAR) úrovni je schopen spustit relaci na S (NUCLEAR), S, C nebo U úrovni. Bezpečnostní úrovně, které jsou přiřazeny k objektům určují citlivost obsahu objektů.

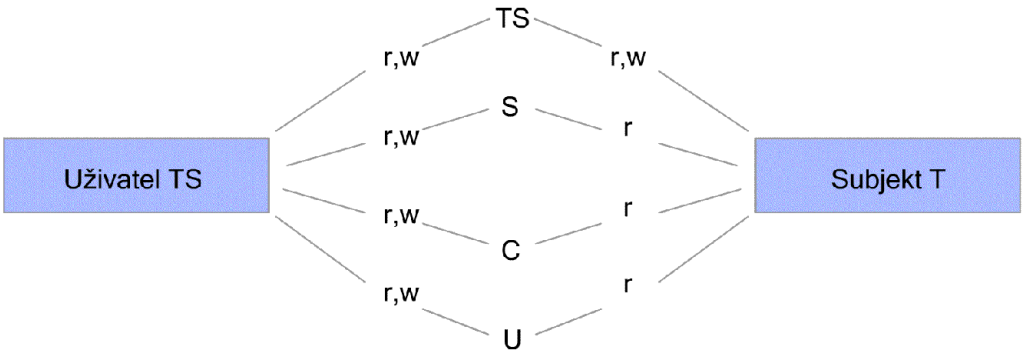
2.4.1 Model Bell-LaPadula

Tento model [2] definuje způsob přístupu v závislosti na dvou vlastnostech:

- *Jednoduchá hvězdičková vlastnost:* Subjekt je oprávněn ke čtení objektu, jestliže bezpečnostní úroveň subjektu převažuje nad bezpečnostní úrovní objektu.
- *Hvězdičková vlastnost:* Subjekt je oprávněn zapisovat do objektu, jestliže bezpečnostní úroveň objektu převažuje nad bezpečnostní úrovní subjektu.

Tyto vlastnosti brání uživatelům od schopnosti číst informaci, která převládá nad jejich autorizační úrovní. Jednoduchá bezpečnostní vlastnost přímo podporuje tuto vlastnost, nikdy nedovolí subjektu číst informaci, která převládá nad uživatelovou bezpečnostní úrovní. Hvězdičkovou vlastnost podporuje režim MAC nepřímo tím, že zabrání subjektům zapisovat informaci úrovně x do kontejneru (obsahu objektu), který by mohl být následovně čtený subjektem s bezpečnostní úrovní, která je ovládána úrovní x. Daná vlastnost tak zabraňuje důležité informaci, aby skončila ve spodním kontejneru, kde si ji může spodní uživatel přečíst.

V tomto modelu je důležité rozlišit uživatele od jeho objektů. Kupříkladu uživatel Michal má přiřazenou úroveň TS (top secret). Michalův rozsah možností zahrnuje schopnost čtení a zápisu u všech objektů, kterým jeho úroveň dominuje. Jeho subjekt na druhou stranu nenabývá takových volností. Ačkoliv můžeme Michalovi věřit, že nevyzradí vysoce tajnou informaci, nejsme schopni poskytnout takovou důvěru Michalovým subjektům, protože subjekty jsou většinou počítačové programy, které mohou být zneužity. Proto, aby Michal úspěšně mohl zapisovat do objektu, např. vysoce tajeného, musí nastavit jeho relaci na úroveň, která je ovládána tajnou bezpečnostní úrovní objektu, viz obrázek níže 2.1.



Obr. 2.1: Uživatelův rozsah schopností versus povolení přístupu subjektu

Mohla by nastat situace, kdy Adam, který je zařazen do důvěryhodné úrovně, by chtěl zcizit tajné nukleární instrukce pomocí trojského koně, tak jako u Petrových

souborů (kap. 2.3). Petr, který je na úrovni S (NUCLEAR), opět spustí Adamův zrádný software, tentokrát během vyvolání S relace (NUCLEAR). Ačkoli tento trojský kůň je schopen číst S (NUCLEAR) instrukce, selže při pokusu zápisu těchto instrukcí na místo, ke kterému má útočník přístup. Je nutno vědět, že Adamův program stále může zničit jakýkoliv Petrův soubor, který je označený v Petrově relaci nebo výše. Jestliže Petr spustí Adamův software během nezařazené relace, Adamův program je schopen zničit veškeré Petrovy soubory.

2.4.2 Bibův model integrity

Tento model [3] byl představen v roce 1977 jako doplněk modelu Bell-LaPadula. Bibův model preferuje spíše integritu dat. V Bibově modelu zákaz čtení a zápisu je založen na úrovních integrity (sestávajících z hierarchických a kategorických komponent), přidělených subjektům a objektům. Integrační úroveň přidělena uživateli určuje úroveň důvěry uživatele co se týče modifikace informace na této úrovni a integrační úroveň spjatá s objektem vyjadřuje citlivost objektu, pokud jde o jeho modifikaci. Například to může obsahovat tyto vlastnosti: kritické (C), důležité (I) nebo běžné (O)[1].

Vlastnosti v tomto modelu jsou stejné jako v modelu Bell-LaPadula až na to, že pravidla kontrolující čtení a zápis jsou opačná. Tyto vztahy jsou popisovány takto:

- *Vlastnost jednoduché integrity:* subjekt má povolení ke čtení objektu, jestliže bezpečnostní úroveň objektu převažuje bezpečnostní úroveň subjektu.
- *Vlastnost hvězdičkové integrity:* subjekt má povolení zápisu do objektu, jestliže bezpečnostní úroveň subjektu převažuje nad bezpečnostní úrovní objektu.

Zápisy z nižší úrovně nejsou povoleny, stejně tak jako čtení z vyšších úrovní směrem k nižším.

2.4.3 Clark-Wilsonův model

Clark a Wilson zpracovali rozdíly mezi požadavky na komerční a vojenskou bezpečnost v roce 1987 [4]. Dospěli k názoru, že pro většinu komerčních aplikací je důležitější integrita než utajení. Integrita v počítačové bezpečnosti se vztahuje k přesnosti a autentičnosti informace. Stejně tak je potřeba se ujistit, že objekty jsou upraveny autorizovaným způsobem od autorizovaného pracovníka.

Clark a Wilson zdokumentovali a zobecnili pohled na vlastnosti komerční integrity tím, že ukázali jak jsou rozdílné od vojensky orientovaných požadavků, které

[1] Tyto zkratky pocházejí z anglických slov: C-critical, I-important, O-ordinary

jsou shrnuty v knize TCSEC. Navrhli dva nejdůležitější principy při ověřování integrity informací: dobře utvořený protokol a SoD[2]. Dobře utvořený protokol omezuje způsoby, jakými uživatel může data upravovat. Proto se ujišťuje, že všechna data, která jsou oprávněně spuštěna, setrvají i po provedení operace. Základní jednotkou řízeného přístupu v Clark-Wilsonově modelu je „trojitá kontorla přístupu“, složená z uživatele, změny procedury a omezené datové položky viz obr. 2.2.

Obr. 2.2: Clark-Wilsonova trojitá kontrola přístupu

Neomezené datové položky (UDIs) jsou ty, které nejsou chráněny modelem integrity. Procedura pro ověření integrity (IVPs) se ujišťuje, zda datová položka se nachází v platném stavu. Clark a Wilson navrhli devět pravidel:

1. Pro každou omezenou datovou položku zde musí být procedura pro ověření integrity, která se ujišťuje, zda datové položky jsou v platném stavu.
2. Procedura přeměny, která mění omezenou datovou položku, musí mít pro tuto změnu správnou kvalifikaci.
3. Omezená datová položka může být změněna jen pomocí pověřené procedury přeměny.
4. Každá procedura přeměny musí provést záznam o změně omezené datové položky.
5. Každá procedura přeměny, která bere vstup z neomezené datové položky musí být kvalifikovaná, aby se ujistila, že jen správným způsobem jsou přeneseny neomezené datové položky do omezených datových položek.
6. Jen kvalifikované procesy přeměny mohou měnit neomezené datové položky.
7. Uživatel může přistupovat k omezeným datovým položkám jen přes procedury přeměny, pro které je uživatel autorizován.
8. Každý uživatel je kontrolován systémem, než spustí proces přeměny.
9. Jen bezpečnostní administrátor je schopen autorizovat uživatele pro proces přeměny.

[2]SoD; Separation of Duty - izolace služby

Bell-LaPadulův bezpečnostní model spoléhá na zprostředkování přístupu v jádru operačního systému, zatímco Clark-Wilsonův model se spoléhá na kontrolu na aplikační úrovni. Tento rozdíl pramení z různých požadavků, které na tyto dva modely byly kladeny. Vojenský víceúrovňový bezpečnostní model kontroluje datový tok, který může být definován termínem nízko-úrovňové operace čtení a zápisu. Komerční model integrity, jak jej definovali Clark a Wilson, se musí ujistit, zda informace je změněna jen autorizovaným způsobem autorizovanými lidmi, a že je nemožné používat kontrolu pouze přes operace probíhající na úrovni jádra. Důležitost kontroly přes transakci může být znázorněna na jednoduché bankovní transakci. Pokladník při ukládání peněz požaduje čtení a zápis do určitých oblastí v rámci souboru určeného pro vklad a v rámci souboru určeného pro záznam transakce. Účetní kontrolor má schopnost provedení opravné transakce a požaduje tak stejný přístup ke čtení a zápisu do stejných souborů jako pokladník. Rozdíl je v procesech, které jsou spuštěny a v hodnotách, které jsou zapsány do souboru pro záznam transakce.

Izolace služby (SoD) je další hlavní část modelu, která přispíva k integritě a předchází autorizovaným uživatelům dělat nevhodné změny. Operace jsou odděleny do vícenásobných podčástí a různé osoby představují určitou podčást. Proces kupování a platby za některé položky může například zahrnovat schválení objednávky, záznam o přijetí položky, záznam o přijetí faktury a schválení platby. Poslední krok by se neměl uskutečnit pokud nebyly splněny tři předchozí kroky. Pokud každý krok představuje jinou osobu, měla by být odhalena a nahlášena neplatná operace. Vyjímkou je spiknutí těchto osob. Pokud každý z těchto kroků provádí jedna osoba, pak je možný podvod - je učiněna objednávka, platba je provedena neexistující společnosti a žádný předmět není doručen. V tomto případě se vše jeví v pořádku, problém však je, že nesedí fyzický stav majetku se soupisem.

2.4.4 Metoda čínské zdi

Brewer a Nash [5] definují tuto metodu jako střet zájmů v souvislosti s bankovnictvím a finančnictvím. Jako u metody Clark-Wilson, metoda čínské zdi je aplikace navrhnuta pro úzkou škálu aktivit, které jsou spjaty s určitými obchodními transakcemi. Cílem čínské zdi je prevence ilegálního toku informací, který má za následek střet zájmů. Poradci mají přirozeně přidělený přístup k majetkovým informacím, aby tak zajistili služby pro své klienty. Když poradce učiní přístup, například ke konkurenční bance, zjistí tak informace, které tak mohou zničit konkurenční výhodu jednoho nebo obou institutů nebo také mohou být zneužity pro osobní prospěch. Úkolem čínské zdi je zjistit a předejít možnému toku informací, který by mohl způsobit tento konflikt.

Citlivé informace společnosti jsou kategorizovány do vzájemně rozdílných kategorií střetu zájmů (COIs). Každá společnost patří jen do jedné této kategorie a každá z nich má dva nebo více členů. Je možné, aby současně existovalo více těchto kategorií. Zatímco COI1 bude náležet bankám, COI2 náleží energetické společnosti. Vlastnost čínské zdi je zaměřena na to, aby poradce nemohl číst informace od více jak jedné společnosti v dané kategorii.

Dokud poradce nepřečetl informaci náležející jakékoliv společnosti, tak není dosud svázán žádným pravidlem a je mu dovoleno číst jakékoliv citlivé informace z jakékoliv společnosti. Když má poradce právo přístupu k citlivé informaci v rámci metody čínské zdi, může mít dále omezen přístup pomocí jiné metody, a to například DAC. Jakmile poradce jednou přečte informaci od banky A, je mu zakázán přístup k informacím ostatních společností patřících do stejné kategorie. Každý poradce má povolení, aby četl veřejné informace všech institucí.

2.4.5 Brewer-Nashův model

Tento model pohlíží na data jako na objekt patřící do souboru dat. Soubory dat společností jsou dále řazeny do COI kategorií.

Brewer-Nashův model definuje dvě pravidla, jedno pro čtení, druhé pro zápis. Co se týče práva čtení, subjekt S je schopen číst objekt O jen pokud je splněno:

- O je ve stejné společnosti souboru dat jako některý objekt předešle čtený S.
- O náleží do COI třídy ve které S již četl objekt.

Pokud jde o právo zápisu, subjekt S je schopen zapisovat objekt O jen pokud je splněno:

- S může číst O podle daného pravidla pro čtení.
- Žádný objekt nemůže být čten v rámci souboru dat různých společností než-li ten, pro který je požadován přístup pro zápis.

Brewer-Nashův model nedělá rozdíly mezi objekty a subjekty, jak je tomu u modelu Bell-LaPadula. Místo toho rozeznává subjekty, aby obsahovaly uživatele a procesy, které jsou vyvolané uživatelem. Stejně tak jako u modelu Bell-LaPadula bere tento model pro právo zápisu v úvahu trojského koně. Za zmínku jistě stojí časový rozdíl mezi těmito modely. Pokud se týče Bell-Lapadula, pravidla pro čtení a zápis mají životnost po dobu relace subjekt-uživatel. U Brewer-Nashového modelu se uživateli právo pro čtení uděluje na celý život. Pokud uživatel přečte objekt ze souboru dat společnosti A, má navždy zabráněný přístup ke čtení objektu z jiného souboru dat společnosti B, patřící do stejné COI kategorie.

2.4.6 DTE model

DTE (domain-type enforcement) mechanismus [6] je tabulkově orientovaný přístup, vynalezený Beobertem a Kainem v roce 1980 jako podpora modelu MAC dle Bell-LaPadula. DTE mechanimus je používán ve firewallech a operačních systémech a ukázal se jako podpora mnoha bezpečnostních pravidel vyjádřených skrz modely RBAC. DTE model jako mnoho dalších modelů kontroly přístupu dělí počítačové systémy do dvou logických entit: subjekty a objekty. Subjekty jsou aktivní entity (většinou procesy). Objekty jsou pasivní entity (např. soubory, adresáře, mechaniky a paměťové prvky). Doména je přiřazena subjektu. Typ je brán jako objekt. Úkol „doménového“ označení pro subjekt záleží na funkci (např. proces obchodní transakce). Objekt je přiřazený typu založeném na požadavku celistvosti. Povolení kontroly přístupu jsou spjaty s doménami i typy. Proto vznikly dvě skupiny povolení: domain-domain povolení a domain-type povolení. Každá z této skupiny povolení je reprezentována tabulkou znaků. Domain-domain přístupová kontrolní tabulka (DDAT) je dvourozměrná tabulka se záznamem pro každý uspořádaný pár domén. Stejně tak domain-type přístupová kontrolní tabulka (DTAT) je dvou-rozměrná se záznamem pro každý pár - doména, typ. Jelikož zde může být více povolení pro páry domain-domain a domain-type, každý záznam v těchto tabulkách obsahuje sadu povolení. Všechny záznamy u těchto dvou typů tabulek společně představují DTE databázi.

- Příklad domain-domain povolení: (C) - create; vytvořit a (K) - kill;vypnout.
- Příklad domain-type povolení: (R) - read; čtení, (W) - write; zápis, (E) - execute; spustit a (T) - prohlížení adresáře.

3 SELINUX

Cílem této kapitoly je seznámit se s SELinuxem, pojmy používaných v jeho souvislosti a s principy vytvaření přístupových pravidel. Jsou zde ukázány příkazy pro klasické ovládání SELinuxu, které jsou pro běženého uživatele složité. Praktická část je tedy věnována SELinux Policy Editoru, který je založen na zjednodušených pravidlech a má grafické rozhraní.

3.1 Co je SELinux

SELinux neboli Security-Enhanced Linux nabízí pro systém větší bezpečnost. Jde o implementaci povinného řízeného přístupu - MAC. Uživatel může být přiřazen k předdefinované roli, kde má přístup jen k těm souborům a procesům, které vlastní. Vztah mezi standardní bezpečnostní politikou v Unixovém systému a SELinuxem je takový, že pokud dojde k určité operaci v systému, jsou nejprve zkontrolována Unixová práva. Po schválení operace přijde na řadu SELinux, který buď operaci povolí nebo zakáže. Pokud však Unix neudělí povolení k operaci, je požadavek ukončen a SE Linux nepřijde ani na řadu. SELinux je založen na několika bezpečnostních modelech: Type Enforcement (TE), RBAC a Multilevel Security (MLS). SELinuxem jsou omezovány pouze operace procházející přes jádro.

3.2 Historie SELinuxu

SELinux pochází z myšlenky vzniklé před několika desetiletími. V roce 1973 počítačový vědec David Bell a Leopard LaPadula definovali koncept bezpečnostního systému a publikovali formální model popisující vícevrstvou bezpečnost systému. Později v osmdesátých letech práci Bella a LaPadula silně ovlivnil TCSEC (oranžová kniha, viz kapitola 2.1). TCSEC definoval šest vyhodnocujících tříd s postupně přísnými bezpečnostními požadavky: C1, C2, B1, B2, B3 a A1. Třída systémů C1 a C2, jako Linux, spadají pod DAC přístup. Třída systémů B1 a systémů vyšších tříd, jako SELinux, museli zahrnovat MAC přístup. Během devadesátých let výzkumy NSA (National Security Agency) pracovali s SCC (Secure Computing Corporation) na vývoji silné a flexibilní architektury MAC přístupu. Zpočátku se soustředili na teoretické důkazy vlastností a charakteristik architektury. Nakonec za spolupráce s výzkumným týmem na univerzitě v Utahu vyvinuli prototyp architektury zvaný Flask. Později NSA pracovala s Network Associates a MITRE, aby zahrnuli tuto architekturu do operačního systému Linux. Jejich práce byla zveřejněna v prosinci 2000 jako open source produkt.

3.3 Jak SELinux pracuje

SELinux [7] spojuje každý program nebo proces s doménou. Ke každé doméně je přiřazena množina povolení tak, aby pracovala správně a nedělala nic, k čemu není předurčena. Například má doména omezený přístup k souborům a k typům operací, které může s těmito soubory provádět. K upřesnění takových povolení má každý soubor tzv. bezpečnostní kontext. Doména nemůže přistupovat k souborům, které mají jiný bezpečnostní kontext, než pro které má explicitně povolený přístup. Proces, který spustí program, opouští jeho aktuální doménu a přejde do nové domény. Nová doména může mít více nebo méně oprávnění než původní doména. Proto programy mohou spustit další programy, mající více nebo méně oprávnění, než sami mají.

V nynějších verzích je SELinux zahrnut do jádra ve formě LSM modulů, které usnadňují práci při konfiguraci. Množina pravidel je nahrávana do těchto modulů a tyto moduly lze za běhu systému přidávat nebo odstraňovat z jádra. Předchází se tak problému z dřívějších let, kdy politika byla jen v jednom modulu a musel se tak při změně politiky kompilovat celý modul.

Součástí SELinuxu je TE (type enforcement), který zajišťuje, že pravidla podle kterých se řídí domény, jsou vždy brána v úvahu. Další částí SELinuxu je RBAC, který limituje uživatelův přístup k doménám. Například některé domény jsou určeny pro přístup pouze systémovému administrátorovi, zatímco jiné jsou určeny pro přístup jakémukoli uživateli. Definice domén, bezpečnostních kontextů a přechodu jsou v souborech, které se nazývají *policy files* (soubory politky) a mohou být upravovány systémovým administrátorem SELinuxu. Proto bezpečnostní politika SELinuxu je vysoce flexibilní a může podporovat širokou škálu bezpečnostních potřeb.

3.4 Politika

Je to soubor pravidel určujících, která role do jaké domény může vstoupit a které domény mohou přistupovat k jakým typům. Soubory politiky lze nastavit podle toho, jak požadujeme, aby systém pracoval. Základní politika je nastavena tak, aby vše zakazovala. Každá operace musí být explicitně povolena v souboru politiky. Například pro uživatele A lze nastavit politiku pro přístup do rolí `user_r` a `sysadm_r` a uživateli B zpřístupnit jen roli `user_r`. Nebo jedna doména smí vytvářet soubory ve složce `/tmp`, ale jiná k nim přístup nemá. Politika kontroluje to, co programy mohou dělat a jak mezi sebou komunikují - tzn. přístup k souborům, jak se vzájemně sledují a posílají signály.

3.5 Používané termíny v souvislosti s SE Linuxem

3.5.1 Identita

Identita neboli uživatel. Tato část může být považována za způsob jak seskupit role. Uživatelé SELinuxu mohou mít více rolí a v těchto rolích více typů. Obvykle se setkáváme se třemi uživateli v systému.

- `user_u` - standardní uživatel SELinuxu,
- `system_u` - označení pro proces, který nebyl spuštěn uživatelem,
- `root` - je uživatel SELinuxu, který se přihlásí jako „root“.

Identita [9] neznamená totéž jako user id v Unixu. Mohou existovat ve stejném systému, mohou mít dokonce stejné textové zastoupení, ale mají jiný význam. Identity v SELinuxu tvoří část bezpečnostního kontextu a ovlivňují, která doména může být zpřístupněna.

3.5.2 Doména

Každý proces běží pod doménou [9]. Doména je v podstatě seznam akcí, které je schopen proces vykonávat.

3.5.3 Typ

Je přiřazen k objektu a určuje, kdo dostane k tomuto objektu přístup [9]. Definice pro doménu je zhruba stejná, až na to, že domény patří k procesům a typy k objektům, jako například adresáře, složky atd. Podle dohody tato část vždy končí `_t`.

3.5.4 Role

Role [9] určuje, která doména může být použita. Domény, ke kterým uživatel role má schopnost přistupovat, jsou předdefinovány v konfiguračních souborech. Jestliže role není autorizována pro přístup k doméně, pak bude zamítnuta. Pro soubory je role značena vždy `object_r`. Procesy jsou značeny obvykle jako `system_r` nebo `sysadm_r`. Role jsou používány k seskupování bezpečnostních typů. Takže je pak v politice jasné, které typy jsou schopné role spustit. Toto je základem pro RBAC v SELinuxu. Role dle dohody končí `_r`.

3.5.5 MLS

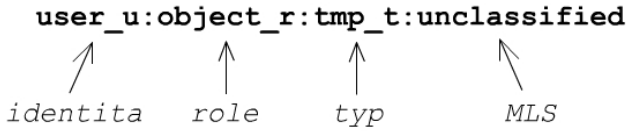
MLS neboli Multilevel Security určuje citlivost dat v rozsahu s0 až s15. Subjekt má právo přistupovat k objektům se stejnou citlivostí. Není podporován v RHEL4 a Fedora Core 4.

3.5.6 MCS

MCS určuje třídu objektu v rozsahu c0 - c255.

3.6 Bezpečnostní kontext

Vše v SELinuxu [8] se vztahuje k bezpečnostnímu kontextu. Jedná se o atributy, které jsou spjaty s položkami jako jsou soubory, adresáře, procesy, TCP sokety apod. Sestává se z identity, role, domény nebo typu a popřípadě obsahuje i MLS. Jednotlivé části jsou od sebe odděleny znakem „:“. Zjištění kontextu, který se vztahuje k souborům lze vypsat příkazem `ls --context` nebo `ls -Z`; pro uživatele `id -Z`; pro proces `ps -Z`. Kontext má tvar `identita:role:doména(typ):mls`. Tento kontext má různá označení, která záleží na tom, k čemu se vztahuje. Pokud se vztahuje k souboru, nazývá se `file_context`. Někdy se mu říká doména, pokud náleží procesu.



Obr. 3.1: Ukázka bezpečnostního kontextu

3.7 Kontrola přístupu pomocí type enforcement

V SELinuxu [8] veškerý přístup musí být udělen explicitně. V základním nastavení SELinux nepovoluje žádný přístup, bezohledu na Linuxovém ID uživatele/skupiny. Znamená to, že zde není žádný standardní superuživatel v SELinuxu. Přístup je udělován pomocí `allow` pravidla. Toto pravidlo má čtyři části.

- *Source type(s)* - obvykle typ domény procesu, který se pokouší o přístup.
- *Target type(s)* - typ objektu, ke kterému přistupuje proces.

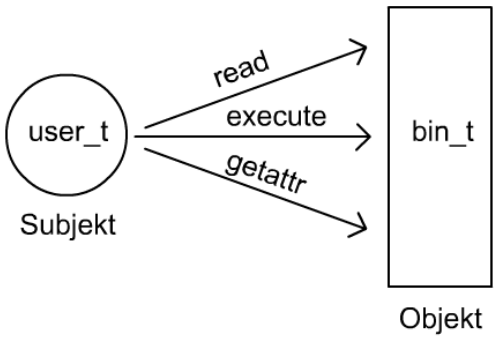
- *Object class(es)* - třída objektu, ke kterému je daný přístup povolen.
- *Permission(s)* - takový přistup, kde zdrojový typ je připuštěn k cílovému typu pro dané třídy objektu.

Příklad `allow` pravidla:

```
allow user_t bin_t : file {read execute getattr};
```

Tento příklad má dva typy identifikátorů: zdrojový *(source)* typ, `user_t`; a cílový *(target)* typ `bin_t`. Identifikátor `file` je název třídy objektu *(object class)* definovaný v politice (v tomto případě běžný soubor). Povolení obsažené v závorkách jsou podmnožinou povolení platných například pro třídu objektu souboru. Přechod tohoto pravidla bude nasledující. Proces s typem domény `user_t` může číst, spustit nebo získat atributy pro objekt souboru s typem `bin_t`.

Povolení v rámci SELinuxu jsou značně jemnější než v Linuxu, ve kterém jsou pouze tři (rwx). V tomto případě, `read` a `execute` jsou doslova jasná, `getattr` je méně zřejmý. Povolení `getattr` k danému souboru povoluje náhled (ne změnu) na atributy jako je datum, čas atd. Ve standardním Linuxovém systému lze tyto informace získat jen s povolením pro vyhledávaní v daném adresáři a to dokonce nemusí mít povolení ke čtení souboru.



Obr. 3.2: Zobrazení pravidla allow

3.8 Typy, atributy a aliasy

Typy [8] jsou základní bloky TE pravidel. SELinux využívá typy k určení povoleného přístupu. Atributy a aliasy jsou rysy potlitiky, které ulehčují správu a používání typů. Atributy jsou používány v souvislosti se skupinami typů s jedním identifikátorem. Aliasy je výhodný mechanismus, který umožňuje definovat alternativní jména

pro typy. Alias identifikátor a identifikátor typu jsou, co se týče politiky, vedeny identicky.

3.8.1 Deklarace typů

Příkaz `type` deklaruje typy a popřípadě alias a atributy tohoto typu. Úplný syntax pro tuto deklaraci vypadá následovně.

```
type type_name [ alias alias_set ] [, attribute_set] ;
```

- `type_name` - identifikátor pro typ, který může mít libovolnou délku a obsahuje znaky ASCII, čísla, podtržítko (_) nebo tečku (.).
- `alias_set` - jeden čí více alias identifikátorů. Má stejné omezení názvu jako identifikátor typu. Pokud je uvedeno více identifikátorů, je mezi nimi mezera a jsou uzavřeny v závorkách, např. `alias {aliasa_t aliasb_t}`.
- `attribute_set` - jeden nebo více předešle deklarovaných identifikátorů atributů. Příklad uvedení více atributů `type bin_t, file_type, exec_type;`.

Příkazy nelze použít v příkazech pro podmínky.

3.8.2 Deklarace atributů

K deklaraci se používá příkaz `attribute`.

```
attribute attribute_name;
```

- `attribute_name` - identifikátor pro atribut. Může mít libovolnou délku a obsahuje znaky ASCII, čísla, podtržítko (_) nebo tečku (.). Atributy jsou na stejném místě jména jako typy a aliasy, a proto nemohou mít stejné jméno jako typ nebo alias. Příkaz nelze použít v příkazu pro podmínku.

3.8.3 Aliasy

Jsou to alternativní jména přiřazovaná k typům. Lze je použít všude tam, kde by se použilo jméno typu, včetně TE pravidel a bezpečnostního kotnextu. Například starší politika se odkazuje na typ `netscape_t`. Aktualizovaná politika by se mohla změnit na typ jména `mozilla_t`, ale poskytuje `netscape_t` jako alias pro povolení starších modulů ke správnému přeložení. Příkaz `typealias` umožňuje definovat alias pro typ.

```
typealias type_name alias alias_names;
```

- `type_name` - jméno typu, ke kterému má být přidán alias. Typ musí být deklarován odděleně, používající `type` příkaz a specifikován může být jen jeden typ.
- `alias_names` - jeden nebo více alias identifikátorů. Mají stejné omezení názvu jako identifikátory typů. Příklad specifikace více typů {`aliasa_t aliasb_t` }

3.9 AV pravidla

Politika [8] SELinuxu podporuje čtyři typy AV pravidel:

- `allow` - povolení přístupu mezi dvěma typy.
- `dontaudit` - zákaz zaznamenavání zpráv o zamítnutém přístupu.
- `auditallow` - povolení zaznamenávání zpráv o povoleném přístupu.
- `neverallow` - specifikuje přístupové povolení, které nikdy nesmí být uděleno pomocí allow pravidla.

Jednoduché AV pravidlo má jeden zdrojový typ, cílový typ, třídu objektu a povolení. Následující příklad má zdrojový typ `user_t`, cílový typ `bin_t`, třídu objektu `file` a povolení `execute`. Toto povolení znamená, že user_t má povolení spustit soubory typu `bin_t`.

```
allow user_t bin_t : file execute;
```

3.9.1 Syntax AV pravidla

```
rule_name  type_set  type_set  : class_set  perm_set;
```

- `rule_name` - jedná se o jméno AV pravidla. Platné položky jsou: `auditallow, auditdeny, dontaudit` nebo `neverallow`.
- `type_set` - jeden či více typů nebo atributů. Je zde oddělený type_set pro zdrojové a cílové typy pravidel. Mnohonásobné typy a atributy jsou určeny seznamem, ve kterých jsou jednotlivé položky odděleny mezerou, např. {`bin_t sbin_t`}. Typy mohou být odděleny od seznamu tím, že na začátek jména typu se napíše pomlčka (např. {`exec_type -sbin_t`}) . Klíčové slovo `self` lze použít v poli cílového typu buď samotné nebo jako součást seznamu typů nebo atributů. `self` nelze použít v poli pro zdrojový typ. Neverallow pravidla také podporují operátor * k obsažení všech typů a operátor ~k obsažení všech typů `except`, kromě explicitně uvedených.

- `class_set` - jeden nebo více objektových tříd. Několikanásobné třídy objektů musí být uzavřeny v závorkách, např. {`file lnk_file`}.
- `perm_set` - jedno nebo více povolení. Všechna povolení musí být platné pro všechny třídy objektů v `class_set`. Více násobná povolení musí být uzavřena v závorkách, např. read, create. Operátor * je využit pro všechna povolení pro všechny třídy objektů. Operátor ~je použit pro určení všech `except` povoleních, kromě explicitně uvedených.

3.10 Role

Příkaz `role` deklaruje identifikátor role a spojuje typy a role. Skladba příkazu pro roli vypadá takto [8]:

```
role role_name [types type_set];
```

- `role_name` - identifikátor pro roli jakékoli délky obsahující ASCII znaky, čísla, tečky a podtržítka.
- `type_set` - jeden nebo více identifikátorů atributů, např. {`user_t passwd_t`}. Příklad vypuštění typu ze seznamu {`exec_type -sbin_t`}.

Pravidlo role `allow` autorizuje změny při spuštění programu.

```
allow role_set role_set;
```

- `role_set` - jeden nebo více identifikátorů. Příklad více identifikátorů {`staff_r sysadm_r`}.

Protože se mohou role při spuštění programu měnit, je třeba zavést pravidla přechodu pro role.

```
role_transition role_set type_set role;
```

- `role_set` - jeden nebo více identifikátorů. Příklad více identifikátorů {`staff_r sysadm_r`}.
- `type_set` - jeden nebo více typů či identifiktorů atributů, např. {`user_t passwdt_t`}. Příklad vypuštění typu ze seznamu {`exec_type -sbin_t`}.
- `role` - nová role pro bezpečnostní kontext.

3.11 Uživatelé

Linuxový atribut identity uživatele není použitelný pro SELinux. Linuxové uživatelské jméno lze kdykoliv změnit a nepodporuje žádnou kontrolu nad dědictvím, postavením nebo inicializací procesu v nové identitě. Mimo jiné superuživatel má právo libovolně linuxové uživatelské identity měnit. SELinux spravuje uživatelův atribut identity v bezpečnostním kontextu, který je nezávislý na Linuxovém atributu identity. SELinux umí pečlivě kontrolovat změny uživatelského atributu bez toho, aniž by ovlivnil kompatibilitu uid Linuxovské sémantiky. Konfigurace politiky limituje schopnost změnit uživatelův atribut identity SELinuxu na určitou TE doménu. Tyto domény jsou spjaty s určitými programy, jako je login, crond a sshd, které byly přizpůsobeny k volání nových funkcí knihovny, aby nastavili správně uživatelovu identitu v SELinux. Z tohoto důvodu uživatelova relace login a cron (nástroj, který spouští různé programy v předem definovanou dobu) jsou ihned spjaty s patřičnou identitou SELinuxu, a proto následné změny uid v Linuxu se neodrazí v rámci uživatelovy identity v SELinux. Zajišťuje se tak uživatelova odpovědnost a vyhýbá se tak bezpečnostním hrozbám.

3.11.1 Deklarace uživatelů

Deklarace uživatele se provádí příkazem `user`, který jej spojí s jednou nebo více rolemi. Tento příkaz se vztahuje jen pro uživatele SELinuxu.

```
user petr roles { user_r };
```

Tento příklad uvádí, že uživatelovi petr je přiřazena role `user_r`. Obecný syntax tohoto příkazu vypadá tedy takto:

```
user user_name roles role_set;
```

- `user_name` - identifikátor uživatele libovolné délky obsahující ASCII znaky, čísla, tečky a podtržítka.
- `role_set` - jeden nebo více identifiktorů, které musí být již v politice předem definovány. Příklad { `staff_r sysadm_r` }.

3.11.2 Mapování uživatelů Linuxu do SELinuxu

```
user user_u roles { user_r };
```

Tento příkaz definuje běžného uživatele `user_u` a autorizuje ho pro roli `user_r`, stejně jako uživatele petr v předchozím příkladě. Rozdíl je v tom, jestli je uživatel `user_u`

definován v politice. Všichni uživatelé Linuxu, kteří nejsou explicitně definováni v politice, jsou namapováni jako `user_u`. Pokud `jana` je identifikátor pro uživatelku Linuxu, ale v politice SELinuxu není uživatelka `jana` definována, tak po přihlášení Linuxového uživatele `jana`, bude identifikátor v počátečním shell procesu `user_u`. Protože petr je definován v politice, jeho počáteční idetnifikátor bude petr.

3.12 Omezení (Constraints)

SELinux [8] poskytuje omezovací mechanismus pro další zamezení přístupu, i když je přístup povolen `allow` pravidlem. Příkaz constrain má tři části: množinu tříd objektů, ke kterým se omezení vztahuje, množinu povolení pro ty třídy, které jsou omezeny a Boolean vyjádření omezení. Syntax příkazu constrain:

```
constrain class_set perm_set expression ;
```

- `class_set` - jedna nebo více tříd objektu. Vícenásobné třídy objektu musí být odděleny mezerou a uzavřeny v závorkách, např. {`file lnk_file`}. Speciální operátory *, ~a - nejsou povoleny.
- `perm_set` - jedno nebo více povolení. Všechna povolení musí být platná pro všechny třídy objektu v class_set. Vícenásobná povolení jsou odděleny mezerou a uzavřená v závorkách, např. {`read create`}. Speciální operátory *, ~a - nejsou povoleny.
- `expression` boolean vyjádření tohoto omezení.

Boolean syntax podporuje tyto klíčová slova:

- t1, r1, u1 Výchozí typ, role, uživatel,
- t2, r2, u2 Cílový typ, role a uživatel.

Syntax omezení také podporuje tyto operátory:

- = = nastaví členství nebo rovnost,
- != není členem nebo neplatí rovnost,
- eq (klíčové slovo jen pro role) rovnost,
- dom (klíčové slovo jen pro role) ovládá,
- domby (klíčové slovo jen pro roli) ovládaný,
- incomp (klíčové slovo jen pro roli) neporovnatelný.

Příklad omezení: `constrain process transition (r1 == r2) ;`
Toto omezení se týká třídy objektu process a omezuje transition (přechodové) povolení pro procesy. Povolení transition je potřebné k umožnění změny domény; toto omezení zakazuje změnu domény. Ve výrazu (u1 = = u2) u1 a u2 určuje zdrojový a cílový uživatelovy identifikátory pro bezpečnostní kontext. Takže tento výraz je pravdivý, pokud zdrojové a cílové uživatelovy identifikátory jsou stejné. V případě změny domény je zdroj aktuální bezpečnostní kontext a cíl je nový bezpečnostní kontext pro proces.

3.13 Konfigurační soubory

Původní konfigurační soubory SELinuxu byly uloženy v adresáři /etc/sysconfig/selinux, který nadále existuje jako symbolický link pro /etc/selinux/config. libselinux čte tento soubor, aby zjistil jak je systém nastaven. Tento soubor obsahuje následující výpis.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
# targeted - Only targeted network daemons are protected.
# strict - Full SELinux protection.
SELINUXTYPE=targeted
# SETLOCALDEFS= Check local definition changes
SETLOCALDEFS=0
```

Lze vidět, že politika SELinuxu je nastavena na targeted a mód je nastaven na enforcing.

3.13.1 Permissive a Enforcing mód

- *Enforcing mód* - zahrnuje všechnu politiku, jako je například zamítnutí přístupu. Jednoduše tento mód dělá to, jak je SE Linux nakonfigurován. Kernel blokuje všechny přístupy až na ty, které nejsou explicitně povoleny. Všechny zamítnuté procesy jsou zaznamenány v logovacím systému jako AVC (Access Vector Cache), pokud však nejsou tyto zprávy explicitně nastaveny, aby se nezaznamenávaly.

- *Permisivní mód* - kernel hlásí porušení přístupu ve formě AVC zpráv, ale přístup povolí. Je pár rozdílů jak kernel hlásí tyto AVC zprávy.

    1. Kernel nahlásí pouze první porušení přístupu v permissive módu pro omezenou doménu na patřičném objektu, kde by však při enforcing módu nahlásil každý zamítnutý přístup.
    2. Může se objevit více dodatečných AVC zpráv, které by se nikdy neukázali v enforcing módu. Například omezená doména měla zákaz čtení adresáře včetně jeho souborů. V enforcing módu přístup do adresáře bude zamítnut a bude vytvořena jedna AVC zpráva. V permissivním módu by přístup do adresáře vytvořil AVC zprávu a každý přístup do souboru by vytvořil AVC zprávu.

Aby se projevily změny v konfiguračním souboru, je nutné systém restartovat. Pro změnu módu se lze tomuto vyhnout pomocí příkazu `setenforce 0` pro zapnutí permisivního módu a pro enforcing mód platí příkaz `setenforce 1`.

3.13.2 Disabled

Pomocí této proměnné zakážeme zcela SELinux v systému. Protože je tato proměnná čtena pouze při bootování systému, je nutné po nastavení na disabled provést reboot systému. Pokud je SELinux vyřazen, je také zamezeno správnému značení souborů. Při zpětném povolení SELinuxu je třeba přeznačkovat celý souborový systém.

3.13.3 SELINUXTYPE

Určuje, ve kterém adresáři libselinux hledá zbytek konfiguračních souborů. Existují tři adresáře /etc/selinux/mls, /etc/selinux/targeted a /etc/selinux/strict, ze kterých si lze vybrat nebo lze vytvořit vlastní.

- strict - politika omezující celý systém, kdy omezení jsou v systémovém i uživatelském prostoru. Při větším množství aplikací by vznikalo mnoho pravidel, které by bylo třeba nakonfigurovat.

- targeted - zabezpečuje pouze služby připojené k síti. V této politice nemá identita a role význam. Uživatel pracuje v doméně `unconfined_t`, která nemá žádné omezení. Procesy mají stejný přístup jako při vypnutém SELinuxu.

- mls - omezení je stanoveno na základě citlivosti a kategorie dat. Tato politika se používá v kombinaci se striktní nebo targeted politikou.

3.14 Soubor .fc

Souborový kontext (file context), který má příponu *.fc*, je umístěn v *file_context/program* podadresář politiky výchozího adresáře. Tento soubor obsahuje tři sloupce.

- Regulární výraz - adresáře a soubory mající cestu shodující se s regulárním výrazem jsou značeny podle specifikace ve sloupcích dva a tři.
- Značky - určují, zda regulární výraz se shoduje s adresáři, soubory nebo s adresáři či soubory.
- Bezpečnostní kontext: Určuje uživatele, roli a typ SELinuxu, kterým je adresář nebo soubor označen.

3.15 Soubor .te

TE soubor (type enforcement file) má příponu *.te* a je umístěn v *domains/program* podadresáři politiky výchozího adresáře. Tento soubor upřesňuje AV (access vector) pravidla a přechody, které jsou spjaty s domény. TE soubor obsahuje:

- Řádek pro typ: Definuje typ a jeho podoba může vypadat takto:

  ```
  type snort_etc_t, file_type, sysadmfile;
  ```

  První část určuje typ, druhá část atribut a třetí část znamená, že souborové objekty mohou být přístupné nebo upraveny pouze uživateli s rolí `system_r` (systémovým administrátorem).
- Řádek pro povolení: Definuje AV pravidlo. Řádek začíná klíčovým slovem `allow`. Např. `allow snort_t etc_t:file { getattr read };` proces běžící v doméně `snort_t` má povolení čtení a získání atributů od složek označených typem `etc_t`.
- Ostatní řádky: Začínají identifikátory jako například `daemon_domain` a `can_network`; volají makra.

3.16 SELinux editor

Jak je vidět v předchozích kapitolách, SELinux je pro běžné uživatele spíše noční můrou. Existuje však SELinux Policy Editor (SEEdit), který konfiguraci SELinuxu výrazně ulehčí. Je založen na tzv. zjednodušené politice a na utilitách, které s touto

politikou umí pracovat. Zjednodušená politika je založena na jazyku SPDL (Simplified Policy Description Language), který řeší obtížnost SELinuxu. Politika SEEditoru je uložena v */etc/seedit/policy*. Pokud je proveden výpis tohoto adresáře, lze vidět, že některé domény už jsou přednastaveny. Mezi tyto domény například patří: init_t.sp, http_t.sp, dhclient_t.sp, crond_t.sp, login_t.sp a další. Originální verze byla vytvořena Hitachi Software, později ji přepsal Yuichi Nakamura.

3.16.1 Přehled grafického editoru

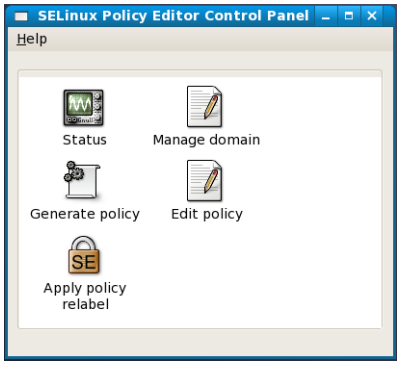
Po otevření SELinux Policy Editoru se zobrazí okno, které obsahuje tyto položky:

- *Status* - nastavuje SELinux mód, zobrazuje běžící procesy a domény pro síťové procesy.
- *Manage Domain* - vytváří novou doménu, odstraňuje nebo zakazuje doménu.
- *Generate Policy* - vytváří politiku z přístupového záznamu.
- *Apply policy, Relabel* - načítá politiku manuálně (editor to však dělá automaticky).
- *Manage Role* - vytváří novou roli; není součástí u starších verzí editoru.

3.16.2 Vytvoření politiky pro aplikaci Pidgin

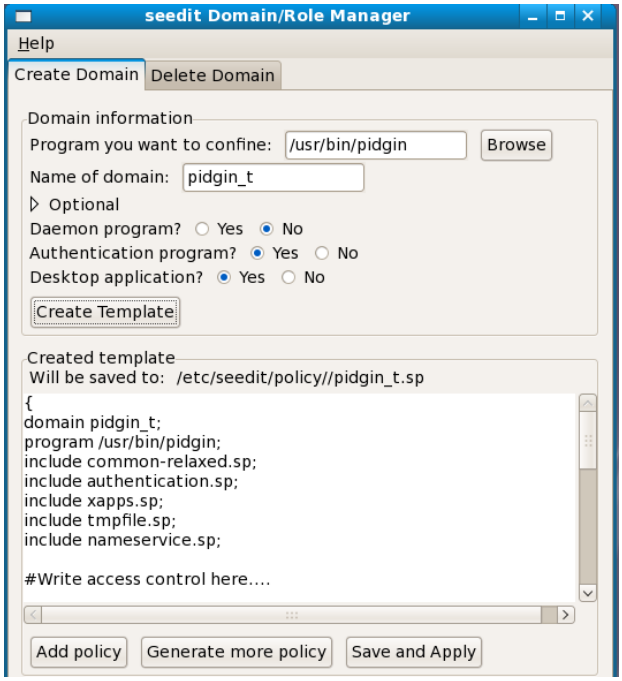
V této kapitole je prakticky ukázáno, jak se vytvoří doména pro aplikaci Pidgin (Internet Messenger).

Nejprve je třeba spustit $Applications -> SystemTools -> SELinuxPolicyEditor$. Pokud není uživatel přihlášen jako `root`, otevře se okno pro přihlášení uživatele `root`. Po úspěšném přihlášení se zobrazí základní nabídka 3.3



Obr. 3.3: Control Panel

Manage domain zobrazí okno 3.4 pro vytvoření nebo odstranění domény. V záložce *Create Domain* je potřeba vyplnit pole *Program you want to confine (program který chceme omezit)*, v tomto případě to je `/usr/bin/pidgin`. Po té se automaticky zobrazí jméno domény `pidgin_t`. Kliknutím na tlačítko *Create Template* se vygeneruje šablona pro řízený přístup. Pomocí tlačítka *Generate more policy* se vytvoří základní pravidla. Tlačítkem *Save and Apply* se tato šablona uloží a potvrdí.



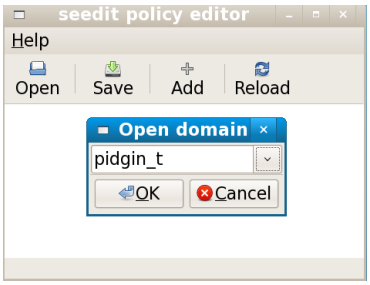
Obr. 3.4: Okno pro vytvoření domény

Nyní po spuštění aplikace Pidgin a následném vytvoření účtu pro ICQ nastává problém s připojením. Je to tím, že není povolen žádný port pro připojení. V okně *Edit policy* 3.5 je třeba otevřít doménu `pidgin_t`. Stisknutím tlačítka *Add* se zobrazí okno 3.6 a v záložce *Network* je nutno vyplnit *Specific Numer: 5190* a zaškrtnout *Behavior; Client*. Změny se potvrdí tlačítkem *add* a v okně *Policy editor* tlačítkem *Save*.

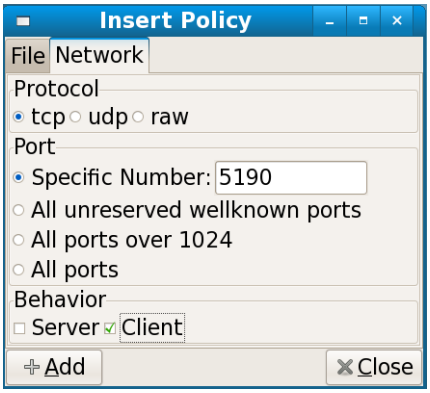
Pidgin se nyní po vytvoření nového účtu je schopen přihlásit. Při novém spuštění však vykazuje další problém, chce totiž znovu vytvořit uživatelský účet. V takovém případě se použije utilita *Generate policy*, která vygeneruje zakázané přístupy. Pak už jen stačí najít správný řádek v záložce *Restult*, zaškrtnout jej a změny potvrdit 3.7.

Jak je vidět na obrázku, zjednodušená politika podporuje tyto atributy:

- - s; povolení vyhledávání v souborovém stromu,



Obr. 3.5: Policy editor



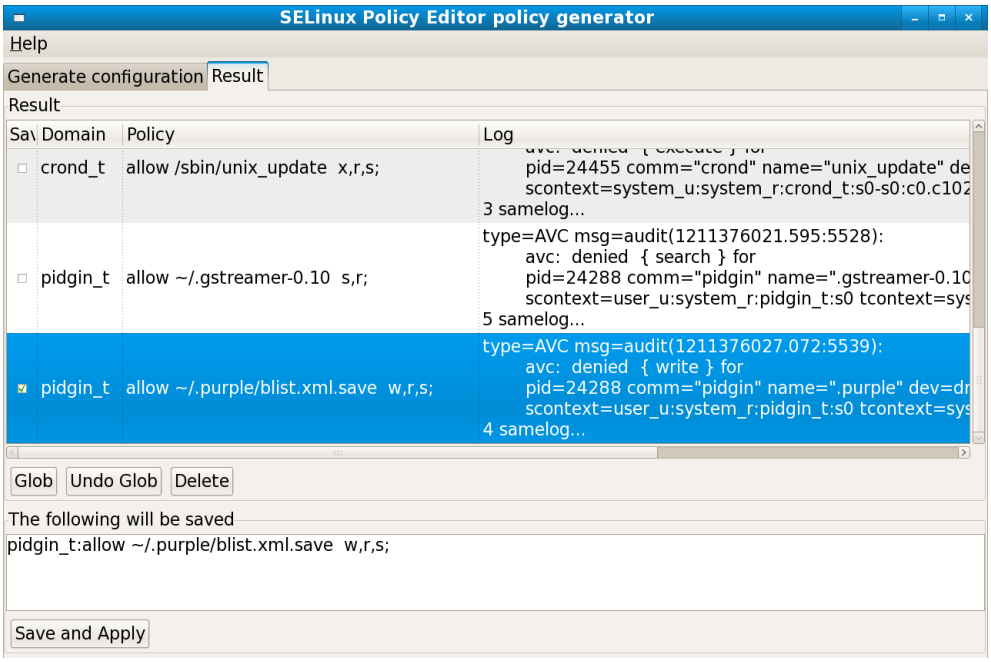
Obr. 3.6: Povolení portu 5190

- - r; čtení souboru,
- - x; spuštění souboru,
- - w; zápis, obsahuje další povoleni: append,create,delete.

Pokud je třeba povolení write blíže upřesnit, lze použít tyto atributy:

- - a; připojení souboru,
- - o; přepsání souboru,
- - c; vytvoření souboru,
- - e; vymazání souboru,
- - t; nastavení atributu souboru (neobsahuje souborový bezpečnostní atribut).

Nyní jsou základní funkce messengeru funkční. Pokud se objeví další zakázané funkce, které by měly být umožněny, použije se k jejich analýze opět utilita *Generate policy*.



Obr. 3.7: Generate policy

3.17 Ovládání SEEdit přes příkazový řádek

Pokud systém nepodporuje grafické rozhraní, je možné nainstalovat SEEdit bez GUI. V takovém případě se konfigurace politiky děje přes příkazový řádek [10].

3.17.1 Zobrazení běžících procesů

```
su
seedit-unconfined -e
```

Na prvním řádku 3.8 se zobrazí aktuální mód SELinuxu. Na dalším řádku lze vidět, že proces init je unconfined (to znamená, že tato doména nemá žádné omezení) a má doménu init_t. Některé z procesů mají omezené chování. Například u procesu sshd je napsáno Confined by sshd_t. Znamená to, že procesu sshd je přidělena doména sshd_t, která je nastavena na omezené chování sshd.

3.17.2 Zobrazení síťových procesů

```
seedit-unconfined -n
```

Ve výpisu 3.9 je zobrazen seznam omezených a neomezených síťových procesů. Je důležité znát stav těchto procesů, protože útoky na systém jsou vedeny především přes síťové procesy.

```
[filip@localhost ~]$ su
Password:
[root@localhost filip]# seedit-unconfined -e
Current SELinux mode: Enforcing

PID     Comm    Domain

1       init    Unconfined(init_t)
2       kthreadd        Unconfined(kernel_t)
3       migration/0     Unconfined(kernel_t)
4       ksoftirqd/0     Unconfined(kernel_t)
5       watchdog/0      Unconfined(kernel_t)
6       events/0        Unconfined(kernel_t)
7       khelper Unconfined(kernel_t)
58      kblockd/0       Unconfined(kernel_t)
61      kacpid  Unconfined(kernel_t)
```

Obr. 3.8: Ukázka výpisu běžících procesů

```
Current SELinux mode: Enforcing

tcp/111 /sbin/rpcbind   Unconfined(initrc_t)
tcp/10000       /usr/bin/perl   Unconfined(initrc_t)
tcp/631 /usr/sbin/cupsd Unconfined(initrc_t)
tcp/25  /usr/sbin/sendmail.sendmail     Confined by sendmail_t
tcp/43967       /sbin/rpc.statd Unconfined(initrc_t)
tcp/22  /usr/sbin/sshd  Confined by sshd_t
udp/32768       /sbin/rpc.statd Unconfined(initrc_t)
udp/32769       /usr/sbin/avahi-daemon  Unconfined(initrc_t)
udp/647 /sbin/rpcbind   Unconfined(initrc_t)
udp/10000       /usr/bin/perl   Unconfined(initrc_t)
udp/703 /sbin/rpc.statd Unconfined(initrc_t)
udp/68  /sbin/dhclient  Confined by dhclient_t
udp/5353        /usr/sbin/avahi-daemon  Unconfined(initrc_t)
udp/111 /sbin/rpcbind   Unconfined(initrc_t)
udp/631 /usr/sbin/cupsd Unconfined(initrc_t)
```

Obr. 3.9: Ukázka výpisu síťových procesů

3.17.3 Přepínaní módu permissive/enforcing

Jsou dva způsoby jak lze změnit aktuální mód SELinuxu. První způsob je pomocí příkazu `setenforce 0` pro permissive mód a `setenforce 1` pro enforcing mód. Druhý způsob je editací souboru */etc/selinux/config* (viz kapitola 3.13.). Aktuální mód zobrazí příkaz `getenforce`.

3.17.4 Vytvoření domény

Ukázka vytvoření nové domény je uvedena pro aplikaci Pidgin, stejně jako v minulé kapitole. Podle uvedeného postupu by měl být uživatel schopen vytvořit jakoukoliv doménu.

Postup pro vytvoření domény je následující:

- Vytvoření předlohy.

- Kontrola domény.
- Otestovat spuštění aplikace a přidání `allow` pravidel do politky.

Příkaz pro vytvoření předlohy domény je následující.

```
seedit-template -d <domain> -e <path to application> -o <output>
```

Konkrétně tedy pro aplikaci Pidgin bude vypadat takto.

```
seedit-template -d pidgin_t -e /usr/bin/pidgin
```

Vygenerovanou předlohu je nutno uložit do souboru */etc/seedit/policy/pidgin_t.sp*. Pomocí *include* jsou v předloze obsažena předefinovaná přístupová práva, obvykle používána pro daemony.

Po vytvoření konfiguračního souboru *pidgin_t.sp* je nutné tuto politiku načíst.

```
seedit-load
```

Další kroky vyžadují přepnutí systém do permissive módu.

```
setenforce 0
```

Pomocí příkazu `seedit-unconfined -e` se ověří, zda Pidgin je vytvořenou doménou omezen. Pokud ano, výpis vypadá takto.

```
PID Comm Domain
13416  pidgin Confined by pidgin_t
```

3.17.5 Nastavení politiky

Po spuštění aplikace Pidgin se zadá do příkazového řádku následující příkaz.

```
audit2spdl -dl
```

Tento příkaz vypíše z *SELinux deny log* zakázané přístupy a navrhne politiku pro danou doménu 3.10. Navržené změny ve formě `allow` povolení je nutno uložit do souboru */etc/seedit/policy/pidgin_t.sp* a pro uplatnění těchto povolení provést `seedit-load`. Nyní je třeba znovu spustit Pidgin a zjistit, zda jsou ještě nějaké přístupy zakázány. Pokud ano, pokračuje se výše uvedeným postupem. Pokud ne, konfigurace je u konce. Je možné, že se aplikace pokouší přistupovat několikrát do stejné složky. Toto pravidlo lze pak zjednodušit pomocí `/**`, tedy například `allow /usr/share/** r,s;` . Další úpravou navrženého pravidla je přístupový port. Vygenerované `allownet` pravidlo totiž povoluje porty od 1024 výše. Pro bežnou komunikaci přes ICQ postačí port 5190, pravidlo proto bude vypadat takto: `allownet -protocol tcp -port 5190 client;` 3.11. Jako poslední věc zbývá zapnout SELinux do enforcing módu.

```
setenforce 1
```

```
-------------------------
#SELinux deny log:
type=AVC msg=audit(1211759705.777:669): avc:  denied  { read } for  pid=4244 comm="pidgin" name="blist.xml"
dev=dm-0 ino=920370 scontext=root:system_r:pidgin_t:s0 tcontext=system_u:object_r:root_dpurple_t:s0 tclass=file
#Suggested configuration
File pidgin_t.sp:
allow /root/.purple/blist.xml  r,s;
-------------------------
```

Obr. 3.10: Ukázka z výpisu zakázaných přístupů

3.17.6 Odstranění politiky

Odstranění politiky se provede odstraněním konfiguračního souboru, který se nachází v */etc/seedit/policy/domainname.sp*. Následující postup přesune konfigurační soubor pouze do adresáře `unused` pro případ, že by bylo potřeba politiku obnovit.

```
# cd /etc/seedit/policy
# mkdir unused
# mv pidgin_t.sp unused
# seedit-load
# /usr/bin/pidgin restart
# seedit-unconfined -e
Current SELinux mode: enforcing
PID Comm Domain
18496 pidgin Unconfined(unconfined_t)
```

Jelikož je konfigurační soubor přesunut do složky unused, lze jej jednoduše znovu aktivovat.

```
# cd /etc/seedit/policy
# mv unused/pidgin_t.sp
# seedit-load
# /usr/bin/pidgin restart
# seedit-unconfined -e
...
```

```
{
domain pidgin_t;
program /usr/bin/pidgin;
include common-relaxed.sp;
include daemon.sp;
include nameservice.sp;

allow initrc_tmp_t  s,r;
allowcom -ipc initrc_t w;
allow unconfined_tmp_t  s,r;
allowpriv ptrace;
allow /root/.purple/prefs.xml  w,r,s;
allow /root/.gstreamer-0.10/registry.i386.xml  r,s;
allow /usr/share/**  r,s;
allow /usr/bin/bug-buddy  x,r,s;
allownet -protocol tcp -port 5190 client;
allow /etc/gtk-2.0/gtkrc  r,s;
allowpriv getsecurity;
allow unconfined_tmp_t  w,r,s;
allow /root/.purple/smileys  s;
allowfs proc_pid_other  r,s;
allowfs proc_pid_other  w,r,s;
allow /usr/bin/gconftool-2  x,r,s;
allowcom -ipc unconfined_t w;
allow /root/.purple/prefs.xml.save  w,r,s;
allow /etc/selinux/config  r,s;
allow gdm_tmp_t  r,s;
allowpriv cap_sys_nice;
allowpriv netlink;
allow /usr/bin/gconftool-2  r,s;
allowcom -ipc xserver_t w;
allowfs proc_pid_other  s;
allow /root/.purple/prefs.xml  r,s;
allow /var/cache/fontconfig/**  w,r,s;
allow /root/.purple/status.xml  r,s;
allow initrc_tmp_t  w,r,s;
allow /usr/bin/gnome-open  x,r,s;
allow /root/.purple/accounts.xml  r,s;
allow /root/.purple/blist.xml  r,s;
allow /etc/fonts/**  r,s;
allow /root/.purple/prefs.xml.save  w,r,s;
}
```

Obr. 3.11: Konečný výpis souboru pidgin_t.sp

4 ZÁVĚR

V dnešní době je ochrana dat realizována pomocí řízeného přístupu. V roce 1983 byl zveřejněn dokument TCSEC, ve kterém jsou definovány dva důležité režimy řízeného přístupu - DAC a MAC.

Režim DAC byl pro armádu nedostatečným zabezpečením tajných dokumentů, protože subjekt s volným přístupek k informaci mohl tuto informaci volně šířit na další subjekt. Další hrozba je útok trojského koně, kterému nedokáže tento režim čelit. Proto byl vyvinut lepší systém kontroly přístupu - MAC.

MAC je víceúrovňové zabezpečení, kdy kontrola přístupu je požadována při každém přístupu k objetu v systému. Pravidla jsou stanovena externě, tudíž uživatelé nemohou dále rozdávat povolení pro přístup k objektům. Proto bezpečnost systému nezávisí na činnosti uživatele. Jak je uvedeno v kapitole 2.4, bezpečnostní úrovně mají hierarchickou a nehierarchickou část. Tyto úrovně jsou přiřazeny k objektům a určují citlivost obsahu objektů.

Režim MAC muže být popisován různými modely.

Model Bell-LaPadula patří výhradně do tajných záležitostí a ignoruje integritu dat.

Jeho doplňkem byl Bibův model představený v roce 1977, který se výhradně zabývá integritou a důvěrností dat.

O deset let později Clark a Wilson dospěli k názoru, že pro komerční potřebu je důležitější integrita než utajení dat. Cílem Clark-Wilsonova modelu je přesnost a autentičnost informace - jen autorizovaným způsobem může autorizovaný pracovník informaci upravit.

Brewer a Nash řadí citlivé informace do rozdílných COI kategorií. Jakmile uživatel přečte informaci v rámci jeho COI kategorie, je mu zakázán přístup k dalším informacím patřících společnostem ve stejné COI kategorii.

Jako poslední byl uveden DTE model specifikovaný Beobertem a Kainem v roce 1980, který slouží jako podpora modelu MAC dle Bell-LaPadula. DTE mechanimus je používán ve firewallech a operačních systémech a ukázal se jako podpora mnoha bezpečnostních pravidel vyjádřených skrz modely RBAC.

Na základě modelu Bell-LaPadula byla vytvořna architektura Flask, která byla později zahrnuta do operačního systému Linux jako SELinux. SELinux přiřazuje programům a procesům domény, které omezují jejich operace, aby byla zaručena bezpečnost systému. Pravidla podle kterých se tyto domény řídí, zajišťuje *type enforcement*. Další složkou SELinuxu je RBAC, který umí omezit přístup uživatele k určité doméně.

Pokud se bude vytvářet politika, je třeba si nejprve uvědomit, jak moc má být systém zabezpečen. Striktní politika omezuje jak uživatelský, tak systémový pro-

stor. Uživatel zde může přistupovat pouze jen k aplikacím a souborům, ke kterým má povolený přístup. Vzniká tak velké množství pravidel, což je velice náročné na konfiguraci. Proto se využívá spíše targeted politika, která omezuje jen místa potencionálního útoku hackerů a důraz je kladen na síťovou bezpečnost.

SELinux je využíván spíše na serverech než na pracovních stanicích. Je to dáno tím, že běžný uživatel je často odrazen časem stráveným nad studiem politiky SELinuxu. V poslední době však vznikají aplikace, jako je například SEEdit, které se snaží politiku SELinuxu zjednodušit a přiblížit tak SELinux i běžnému uživateli.

LITERATURA

[1] FERRAIOLO, David F. , KUHN, D. Richard, CHANDRAMOULI, Ramaswamy. *Role-Based Access Control.* 2nd edition. [s.l.] :Artech House Publishers, 2007. 418 s. ISBN 978-1596931138.

[2] Bell, D. E., and L. J. LaPadula. *Secure Computer Systems: Mathematical Foundations and Model.* Bedford, MA: The Mitre Corporation, 1973.

[3] Biba, K.J. *Integrity Considerations for Secure Computer Systems.* Bedford, MA: The MITRE Corporation, 1977.

[4] Clark, D.D., and D.R.Wilson. *A Comparison of Commercial and Military Computer Security Polices.* IEEE Symposium of Security and Privacy, 1987.

[5] Brewer, D., and M. Nash. *The Chines Wall Security Policy.* Proc. IEEE Computer Society Symposium on Reasearch in Security and Privacy, April 1989.

[6] Badger, L., et al. *A Domain and Type Enforcement Prototype, Usenix Computing Systems*, Volume 9, Cambridge, MA, 1996.

[7] Bill McCarty *SELinux NSA's Open Source Security Enhanced Linux*, O'Reilly, October 2004 ISBN 0-596-00716-7

[8] Frank Mayer, Karl MacMillan, David Caplan *SELinux by Example: Using Security Enhanced Linux*, Prentice Hall, July 27. 2006, ISBN 978-0-13-196369-6

[9] Faye Coker *Getting started with SELinux* [online], Last update 06 December 2003 Dostupný zWWW:
<http://www.linuxtopia.org/online_books/getting_started_with_SELinux>.

[10] Yuichi Nakamura *SELinux Policy Editor(SEEdit) Administration Guide 2.1*, Dostupný zWWW:
<http://seedit.sourceforge.net/doc/2.1/tutorial.pdf>.

5 SEZNAM ZKRATEK

RBAC přístupová kontrola založena na rolích - Role Based Access Control

CIA utajení, integrita, dostupnost - Confidentiality Integrity Availability

DAC volná kontrola přístupu - Discretionary Access Control

MAC povinná kontrola přístupu - Mandatory Access Control

TCSEC dokument definující standardy MAC a DAC - Trusted Computer System Evaluation Criteria

ACL seznam povolených přítupů - Access Control List

SoD izolace služby - Separation of Duty

TP procedura přeměny - Transformation Procedure

CDI omezená datová položka - Constrained Data Item

UDI neomezená datová položka - Unconstrained Data Item

IVPs procedury pro ověření integrity - Integrity Verification Procedures

COIs kategorie střetu zájmu - Conflict-Of-Interest categories

DTE porsazení domén a typů - Domain and Type Enforcement

SELinux - Security-Enhanced Linux

TE prosazení typů - Type Enforcement

MLS - vícevrstvá bezpečnost - Multilevel Security

SEEdit - editor pro SELinux - SELinux policy Editor

GUI - grafické uživatelské prostředí - Graphical User Interface

LSM - Linux Security Module