Міністерство освіти і науки України
Державний вищий навчальний заклад
«Національний гірничий університет»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
дипломної роботи

магістра
(ступінь підготовки)

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

напрям підготовки (спеціальність) *125 Кібербезпека*
(код і назва напряму підготовки)

спеціалізація (освітня програма) *Кібербезпека*
(код і назва спеціальності)

ступінь підготовки *Магістр*
(назва освітнього рівня)

кваліфікація *професіонал із організації інформаційної безпеки*
(код і назва кваліфікації)

на тему: ***Оцінка механізмів мережевої безпеки на основі політики RBAC***

Виконавець: студент *2* **курсу, групи** ***125м-16-1***

_____ ***Безкоровайний Валерій Віталійович***
(підпис) (прізвище ім'я по-батькові)

Керівники	Прізвище, ініціали	Оцінка	Підпис
роботи	д.т.н., проф. Корнієнко В.І.		
розділів:			
спеціальний	ст.викл. Святошенко В. О.		
економічний	к.е.н., доц. Волотковська Ю.О.		

Рецензент			

Нормоконтроль	ас. Мєшков В.І.		

Дніпро
2018

Міністерство освіти і науки України
Державний вищий навчальний заклад
«Національний гірничий університет»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

«_____»_____20___року

ЗАВДАННЯ
на виконання кваліфікаційної роботи магістра

напряму підготовки (спеціальності) *125 Кібербезпека*
(код і назва спеціальності)

студенту ***125м-16-1*** ***Безкоровайному Валерію Віталійовичу***
(група) (прізвище ім'я по-батькові)

Тема дипломної роботи ***Оцінка механізмів мережевої безпеки на основі політики RBAC***

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора Державного ВНЗ «НГУ» від _____ № _____

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень ***Процес управління доступом до ресурсів інформаційної системи***

Предмет досліджень ***Розмежування доступу до ресурсів на основі моделі RBAC***

Мета НДР ***Оцінка механізмів мережевої безпеки та впровадження моделі RBAC для вирішення питання розподілу рольового доступу при великій кількості співробітників***

Вихідні дані для проведення роботи ***матеріали науково-дослідної та переддипломної практик***

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна ***Наукова новизна полягає в тому, що дана тема піднімає питання, щодо покращення захисту мережевої безпеки та зменшення навантаження на мережу та на системного адміністратора, тому підвищується якість та цілісність мережі***

Практична цінність ***Значно покращується розподіл контролю дозволу, завдяки присвоєнню ролей та розподіленню користувачів на групи, які мають свій рівень доступу***

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Проаналізувати мережеву безпеку на основі політики RBAC, вирішити питання розподілу ролей серед користувачів та зменшення ризиків навмисної шкоди співробітників, які мають рольовий доступ.

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Огляд джерел за темою та напрям досліджень	13.10.17-06.11.17
Методи досліджень	15.10.17-27.11.17
Результати досліджень	30.11.17-24.12.17
Виконання економічного розділу	19.12.17-29.12.17
Оформлення пояснювальної записки	08.01.18-15.01.18

6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект ***Доцільність витрат на використання мережевої безпеки на основі наданої політики та зменшення збитків***

Соціальний ефект ***Чіткий розподіл ролей серед користувачів, зменшення ризику зловживанням посадою співробітника***

7 ДОДАТКОВІ ВИМОГИ

Оформлення роботи повинно відповідати ДСТУ 3008-95 «Документація. Звіти у сфері науки і техніки. Структура і правила оформлення»

Завдання видав _____ Святошенко В. О.
(підпис) (прізвище, ініціали)

Завдання прийняв до виконання _____ Безкоровайний В. В.
(підпис) (прізвище, ініціали)

Дата видачі завдання: 01.09.17р.
Термін подання дипломної роботи до ДЕК 16.01.18р.

РЕФЕРАТ

Пояснювальна записка: 87 с., 24 рис., 3 табл., 3 додатки, 54 джерел.

Об’єкт дослідження: процес управління доступом до ресурсів інформаційної системи.

Мета роботи: оцінка механізмів мережевої безпеки та впровадження моделі RBAC для вирішення питання розподілу рольового доступу при великій кількості користувачів.

Методи дослідження: методи індукції, аналізу і синтезу (при розкритті теоретичних положень); метод обробки інформації (при розрахунках параметрів).

У спеціальній частині дана характеристика моделі RBAC, поняття про мережеву безпеку та описуються проблеми які виникають при контролі доступу серед користувачів.

В економічному розділі визначені витрати на дослідження та розробку моделі GB-RBAC та у порівнянні доведено доцільність цих витрат зі зменшенням збитку.

Практичне значення роботи полягає у застосуванні моделі GB-RBAC та її переваги перед іншими. Проведено аналіз доцільності моделі для вирішення поставлених задач.

Наукова новизна дослідження полягає у покращенні захисту мережевої безпеки та зменшення навантаження на мережу та на системного адміністратора, тому підвищується якість та цілісність мережі.

МЕРЕЖЕВА БЕЗПЕКА, ПОЛІТИКА БЕЗПЕКИ, RBAC, GB-RBAC, РОЛЬОВИЙ ДОСТУП, ГРУПИ РОЛЕЙ ТА ДОЗВОЛІВ, ВІРТУАЛЬНА ГРУПА, КОЛЬОРОВІ МЕРЕЖІ ПЕТРІ, ФУНКЦІОНАЛЬНІСТЬ КІНЦЕВОГО ПОТОКУ, ФУНКЦІОНАЛЬНІСТЬ КАНАЛУ, ФУНКЦІОНАЛЬНІСТЬ ПЕРЕТВОРЕННЯ, РОЛЬОВІ КОНФЛІКТИ

РЕФЕРАТ

Пояснительная записка: 87 с., 24 рис., 3 табл., 3 приложения, 54 источник.

Объект исследования: процесс управления доступом к ресурсам информационной системы.

Цель работы: оценка механизмов сетевой безопасности и внедрение модели RBAC для решения вопроса распределения сетевого доступа при большом количестве пользователей.

Методы исследования: индукции, анализа и синтеза (при раскрытии теоретических положений), обработки информации (при расчетах параметров).

В специальной части дана характеристика моделей RBAC, понятие про сетевую безопасность и описываются проблемы, которые возникают при контроле доступа среди пользователей.

В экономическом разделы определенны затраты на исследования и разработку модели GB-RBAC и, в сравнении, доказано уместность расходов для уменьшения убытка.

Практическое задание работы состоит в применении модели GB-RBAC и ее преимущества среди прочих. Проведен анализ уместности модели для решения поставленных задач.

Научная новизна исследования заключается в улучшении защиты сетевой безопасности и уменьшении нагрузки на сеть и на системного администратора, поэтому улучшается качество и целостность сети.

СЕТЕВАЯ БЕЗОПАСНОСТЬ, ПОЛИТИКА БЕЗОПАСНОСТИ, RBAC, GB-RBAC, РОЛЕВОЙ ДОСПУТ, ГРУППЫ РОЛЕЙ И РАЗРЕШЕНИЙ, ВИРТУАЛЬНАЯ ГРУППА, ЦВЕТНЫЕ СЕТИ ПЕТРИ, ФУНКЦИОНАЛЬНОСТЬ КОНЕЧНОГО ПОТОКА, ФУНКЦИОНАЛЬНОСТЬ КАНАЛА, ФУНКЦИОНАЛЬНОСТЬ ПРЕВРАЩЕНИЯ, РОЛЕВЫЕ КОНФЛИКТЫ.

ABSTRACT

Explanatory note: 87 p., 24 figures, 3 tables, 3 appendixes, 54 sources.

Object of research: the process of managing access to information system resources.

The purpose of the work: assessment of network security mechanisms and implementation of the RBAC model to solve the issue of distribution of network access with a large number of users.

Methods of research: induction, analysis and synthesis (at the disclosure of theoretical positions), information processing (in the calculation of parameters).

The special section features RBAC models, the notion of network security, and describes the problems that arise when controlling access among users.

In the economic section, the cost of exploring and developing the GB-RBAC model is determined and, in comparison, cost-effectiveness is proven to reduce the loss.

The practical task of the job is to apply the GB-RBAC model and its benefits among others. An analysis of the appropriateness of the model for solving the problems has been carried out.

The scientific novelty of the research is to improve the protection of network security and reduce the load on the network and the system administrator, which improves the quality and integrity of the network.

NETWORK SECURITY, SECURITY POLICY, RBAC, GB-RBAC, ROLE-ACCESS, ROLE GROUPS AND PERMISSIONS, VIRTUAL GROUP, COLOR PETRI NETWORK, END-FLOW FUNCTIONALITY, CHANNEL FUNCTIONALITY, TRANSFORM FUNCTIONALITY, ROLE CONFLICTS.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

AEF – Active End-Flow;

CPN – Color Petri Network;

DAC – Discretionary Access Control;

DMZ – Demilitarized Zone;

EF – End-Flow;

FTP – File Transfer Protocol;

GB-RBAC – Group-Based Role-Based Access Control;

GTRBAC – Generalized Temporal Role-Based Access Control;

RBAC – Role-Based Access Control;

MAC – Mandatory Access Control;

NIST – National Institute of Standards and Technologies;

PN – Petri Network;

PEF – Passive End-Flow;

SSO – Single Sign-On;

VG – Virtual Group;

XACML – eXtensible Access Control Markup Language.

ЗМІСТ

ВСТУП

Протягом останніх десятиліть спостерігається поява та широке використання систем електронної комерції та електронного керування, в яких веб- та Інтернет-послуги стають всеосяжними.

З ростом потреб розмежування дозволу до інформаційних ресурсів серед користувачів – розвиваються моделі, механізми інформаційної безпеки. А саме, росте потрібність у мережевій безпеці.

Останнім часом збільшується кількість монопольних компаній, де кількість співробітників тисячі людей. За для забезпечення мережевої безпеки всередині компанії необхідно налаштувати внутрішню мережу та підтримувати її розвиток та контроль з ростом кількості співробітників та нових користувачів.

Щоб бути впевненим у цілісності інформації та зменшити ризик витоку інформації через співробітника, було розроблено моделі контролю доступу на основі розподілення ролей серед користувачів. Кожному користувачеві відводиться своя роль, а отже і обмежена інформація.

Та зі збільшенням кількості користувачів виникають нові проблеми з контролем та розподіленням ролей. Адміністратор фізично не вправі оброблять велику кількість інформації та прослідкувати все.

Тому було представлено модель групової концепції на основі моделі RBAC під назвою GB-RBAC. Яка, в свою чергу, вирішує безліч питань для мережевої безпеки з великою кількістю користувачів.

РОЗДІЛ 1

ЗАГАЛЬНІ ВІДОМОСТІ ПРО МЕРЕЖЕВУ БЕСПЕКУ

1.1 Стан питання

1.1.1 Актуальність питання

Протягом останнього десятиліття спостерігається поява та широке використання систем електронної комерції та електронного керування, в яких веб- та Інтернет-послуги стають всеосяжними. Все більше і більше спільних програм було розроблено на основі існуючих систем та інфраструктур для підвищення ефективності та продуктивності. Як правило, у спільній роботі користувачі різних груп співпрацюють, керуючись деякими спільними ресурсами. Зі зростанням масштабованості співпраці, контроль над доступом та використанням інформації та послуг стає дуже складним.

1.1.2 Проблема безпеки в співпраці

Проектування, експлуатація та обслуговування конфігурації мережі складають важливу частину завдання управління безпекою. Безпека розподілених додатків підтримується набором служб мережевої безпеки, які реалізуються за допомогою механізмів безпеки.

Адміністратор повинен визначити служби безпеки для використання, а конфігурації механізмів безпеки для застосування. Після застосування, політика безпеки мережі часто стає некерованою з плином часу, оскільки додаються додаткові правила і існує реальна трудність в отриманні, управлінні і позбавленні від старих непотрібних правил. Цей факт призводить все до більшого посилення управління активними політиками безпеки в мережевих пристроях.

Основна проблема безпеки в співпраці полягає в тому, щоб контролювати допуск користувачів до співпраці та їх дозволи на доступ до ресурсів, зазвичай, на основі їх обов'язків чи вмінь. Дослідники запропонували і впровадили багато

моделей контролю доступу. Серед них Role-Based Access Control (RBAC) – це найбільш привабливе рішення.

1.1.3 Загальні відомості про RBAC модель

В RBAC-моделі, як дозволи, так і користувачі призначаються ролями системних адміністраторів, таким чином, що користувач отримує дозволи для призначених ролей.

Багато вдосконалених моделей RBAC було запропоновано для авторизації в багатодоменних середовищах. Наприклад, запропоновано структуру політики, для аналізу політики інтеграції RBAC-політики з декількох доменів та розподілу обмежень робочої сили. Більшість із цих останніх робіт спрямовані на безпечне співробітництво, використовуючи безпосереднє картографування ролей і може вирішити деякі проблеми при інтеграції політики RBAC, таких як просування ролей та порушення специфічного розподілу обов'язків, ролевого характеру та порушень щодо ролі. Проте в підході картографічної ролі, роль у одній групі відображається в ролі іншої групи, яка вимагає нетривіальних зусиль адміністраторів безпеки для управління дозволами, особливо з сотнями або тисячами користувачів, ролями та їх взаємозв'язками, оскільки в загалом, лише невелика команда адміністраторів безпеки делегована для керування цими компонентами.

1.1.4 Ефективність RBAC

Проблема стає ще гіршою при співпраці, де потрібні динамічні ролі та дозвіл-ролі. Наприклад, для різних відео-конференцій потрібно, щоб учасники брали участь у різних групах різних користувачів, тому для адміністраторів повинно бути багато адміністративних завдань для динамічної зміни ролей і дозволів для відео-конференцій. Це неможливо і нездійсненне для деяких місцевих адміністраторів виконувати ці завдання. Тому питання управління RBAC є важливим фактором, який прямо обмежує його розгортання та використання в динамічному співробітництві.

Поняття елемента керування доступом на основі ролі (RBAC) розпочалося з багатокористувацьких та багатопрограмних онлайнових систем і приймається як більш просунутий метод контролю доступу, ніж існуючими: дискреційний контроль доступу (DAC) або обов'язковий контроль доступу (MAC). У моделі RBAC дозволи призначаються для ролей, а також користувачам призначаються відповідні ролі, і таке поняття дозволяє менеджеру безпеки ефективно керувати численними дозволами.

1.1.5 Класифікація моделі RBAC

Навчання NIST (Національний інститут стандартів і технологій) вказує на те, що дозволи, призначені для ролей, як правило, змінюються відносно повільно, порівняно зі змінами членства в ролі користувачів. У дослідженні також було встановлено, що бажано дозволити адміністраторам надавати та відкликати членство для користувачів у існуючих ролях, не даючи повноваження цим адміністраторам створювати нові ролі або змінювати призначення завдання на роль. З цієї причини багато попередніх досліджень, що стосуються моделей RBAC в першу чергу зосереджується на взаємозв'язку між користувачами та ролями. Тим часом, дослідження стандартів RBAC NIST класифікували моделі RBAC на плоскі RBAC, ієрархічні RBAC, обмежені RBAC та симетричні моделі RBAC.

Серед моделей функції для розгляду ролевих відносин дозволів були додані до симетричної моделі RBAC, але вони не були конкретно зрозумілими до теперішнього часу. Причиною є те, що дуже важко визначити єдину симетричну модель RBAC для дозволів, оскільки кожен з них має різні характеристики залежно від областей, до яких застосовується RBAC. Тим не менше, збереження належних та точних дозволів на отримання дозволів є важливим компонентом будь-якої схеми управління авторизацією. У випадку, якщо для ролі призначаються невідповідні дозволи, то призначення користувачів цій ролі стає абсолютно безглуздим. Отже,

симетрична модель RBAC, яка дає відповідні дозволи для кожної ролі, обов'язково потрібна.

1.1.6 Управління на основі політики

Традиційні платформи управління, занадто прості для вирішення проблеми складності.

Потрібно знайти підхід до управління на основі політики, яка враховує абстрактні політики безпеки, які можуть бути представлені на різних рівнях, мають діапазон від бізнес-цілей до змін властивостей, специфічних для пристрою. Процес, який перетворює певну мету в відповідні конфігурації, називається процесом деривації. З аналогічною перспективою, заснована багатоагентна системна парадигма, щоб агенти управління були автономніше, щоб мати можливість співпрацювати для створення стратегій, які відповідають певним цілям. Терміни, стратегії цілі, які використовуються тут, після рівня абстракції політики в колишньому підході. В якості третього підходу, останні новітні роботи приступили до цієї ідеї за допомогою концепції «Само-адаптивних автономних обчислень».

1.1.7 Метод оцінки для визначення правильної оцінки мережевої безпеки

Однак автоматизація є поки що недостатньою для управління безпекою. Для автоматичної оцінки немає методів політики мережевої безпеки. Моделі контролю доступу забезпечують рішення для визначення цілей безпеки.

Фактично вони надають формальну методику визначення того, що дозволено і що не дозволено. Крім того, існує ще кілька методів, пов'язаних з кожною моделлю, щоб гарантувати правильність політики безпеки.

Проте, ці моделі не розглядаються як об'єднані механізми безпеки або стратегії. Управління безпекою мережі, за своєю природою, є розподілом функціональності, яка забезпечує узгодження різних пристроїв з різними можливостями (ПК, маршрутизатори, захищені шлюзи, брандмауери і т. Д.).

Внаслідок цього та ж мета може бути застосована різними структурами, таким чином, різними стратегіями.

Наприклад, конфіденційність може бути реалізована за допомогою фільтруючих механізмів або механізмів шифрування. Потім необхідно розробити автоматизований формальний метод оцінки для визначення правильної стратегії мережевої безпеки.

Існує безліч формальних методів перевірки, використовуваних в контексті забезпечення безпеки: теоретичні провізори (EHDM, PVS) і перевірка/виявлення моделі (SMV, NPA, Alloy). Всі мови офіційних специфікацій, такі як Z, LOTOS або Petri Nets - також використовуються. На жаль, немає моделі, пов'язаної з мережевою безпекою, пропонованої для використання з цими методами. Відповідно, пропонується новий інструмент перевірки, який є унікальним для політики мережевої безпеки. Він включає в себе модель додатку політики безпеки, політики і механізмів мережевої безпеки.

1.2 Визначення політики мережевої безпеки

Серед моделей контролю доступу, обрали NIST RBAC модель, оскільки вона спрощує завдання управління. Власне, концепція ролі дозволяє агрегувати дозволу користувачів, а потім спрощує роботу користувачів, зміни прав, внесені адміністратором. Більш того, ієрархії між ролями є хороший інструмент для моделювання організації відповідно до різних точок зору.

1.2.1 Модель NIST RBAC

Група NIST пропонує стандартизацію моделі RBAC. Вона складається з двох під-моделей: основної моделі і ієрархічної моделі (рисунок 1.1). Основна модель включає п'ять наборів базових елементів даних:

• «Користувач» є активним об'єктом, тобто людським або інтелектуальним агентом.

• «Роль» - це функція роботи в контексті організації з деякою пов'язаної семантикою щодо повноважень і відповідальності користувача призначеної ролі. Ми можемо помітити, що це визначення не конкретне.

• «Дозвіл» - це дозвіл на виконання операції по одному або декільком захищених об'єктів.

• «Операція» - це виконуваний образ програми, який після виклику виконує певну функцію від імені користувача.

• «Об'єкт» - це об'єкт, який містить або отримує інформацію.

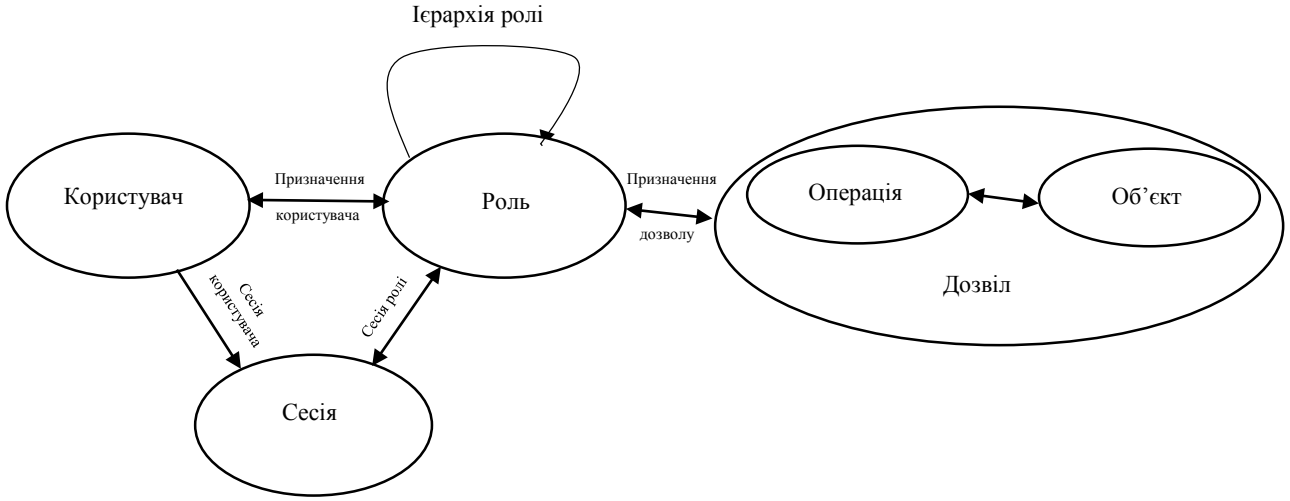

Рисунок 1.1 – Модель NIST RBAC

Користувачеві призначається набір ролей і призначається набір дозволів до ролі. Сесія – це зіставлення одного користувача з набором дозволених ролей. Ієрархічна модель додає відносини для підтримки ієрархій ролі. Існують різні підходи до побудови ієрархії ролей: на основі привілеїв або на основі функцій роботи користувачів.

1.2.2 Зв'язок між політикою безпеки додатку і мережею політики безпеки.

Коли користувач звертається до служби, здійснюється обмін даними між пристроєм, на якому користувач запускає службу і пристроєм, що підтримує послугу (рисунок 1.2). Таким чином, зв'язок, між політикою мережевої безпеки і політикою безпеки додатків, може бути сприйнята. Наприклад, якщо політика

безпеки додатку стверджує, що користувач u1 може читати об'єкт o1 - зазначимо (u1, o1, + read), це означає, що відповідний потік даних (o1, + read) між пристроєм користувача u1 і пристроєм o1 може існувати в мережі . Отже, пов'язана політика мережевої безпеки повинна забезпечувати потік даних (o1, + read) між цими двома пристроями - зазначено (пристрій (u1) ↔ пристрій (o1), + потік (o1, read)).

І навпаки, якщо в політиці безпеки програми зазначено, що користувач u2 не може читати об'єкт o2 (u2, o2, - read), то не повинно бути потоку (o2, read) між пристроями u2 і o2. Тому політика мережевої безпеки повинна заборонити потік (o2, read) між пристроями u2 і o2, тобто (пристрій (u2) ↔ пристрій (o2), - потік (o2, read)). Ми повідомляємо цю інформацію з програми мережевого рівня, щоб зупинити ці потоки даних і таким чином запобігти забороні атаки служб або експлойтів/корисних навантажень.

Визначення 2.1. Відносне ставлення виведення ⇒d визначається як ∀u ∈ USERS, ∀o ∈ OBJECT S, ∀a ∈ ACTIONS, (u, o, ± a) ⇒d (пристрій (u) ↔ пристрій (o), потік (o, ± a)).

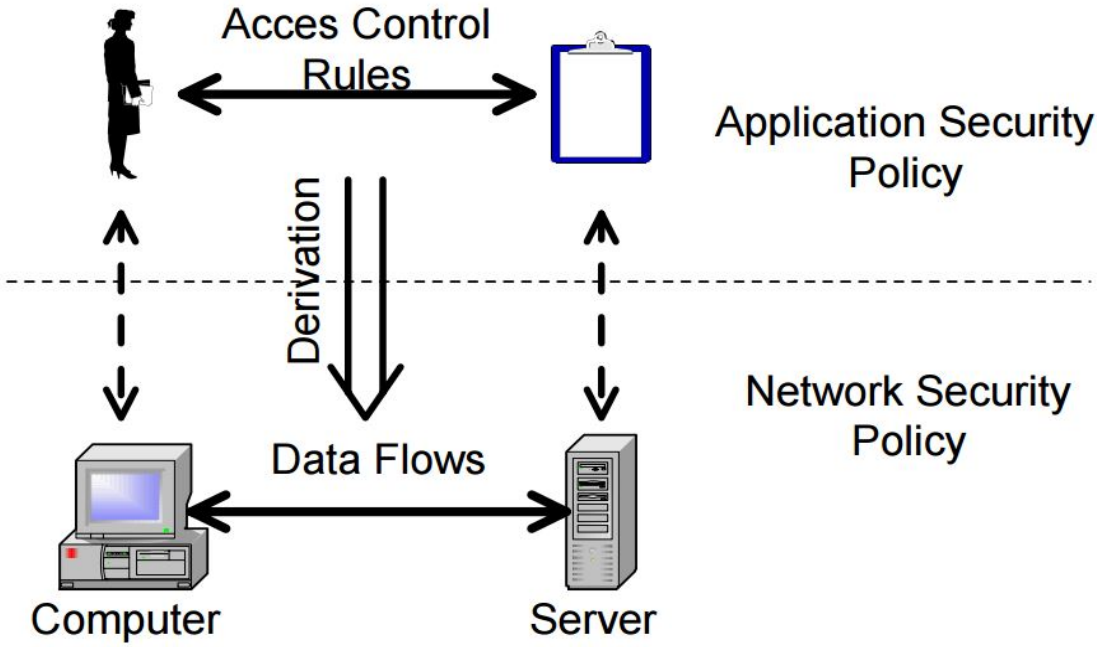

Рисунок 1.2 – Виведення політики безпеки

1.2.3 Політика безпеки мережі RBAC

Надалі будемо вважати, що немає ієрархії і, що ролі не перетинаються (якщо це не так, потрібно створити розділ цього набору): таке обмеження допоможе об'єднати потоки даних на основі призначених дозволів до однієї ролі, а потім ідентифікувати їх за роллю. Згодом відзначається ім'я ролі: набір потоків, відповідний дозволами, присвоєним ролі.

Відповідно до цих визначень представляється метод, який включає в себе мережу специфікацій архітектури та валідацію механізмів безпеки проти процесу політики безпеки RBAC.

1.3 Модель мережевої архітектури та аналіз механізмів безпеки

В мережевому середовищі, згідно з усіма методами обробки потоку даних, можна виділити чотири категорії функціональних можливостей:

• механізми, які споживають/виробляють потоки даних, такі як кінцеві системи;

• механізми, які поширюють потоки даних, такі як опори зв'язку;

• механізми, які перетворюють потоки даних в інші, такі як протоколи безпеки;

• механізми, які фільтрують потоки даних, такі як брандмауери.

Процес полягає в моделюванні цих функціональних можливостей і взаємодій між цими функціональними можливостями. Визначається графічна мова з формальної семантикою, щоб підтримувати процес перевірки політики мережевої безпеки.

У моделі (рисунок 1.3) знаходиться набір активних об'єктів, набір пасивних об'єктів і набір функціональних можливостей (кінцевий потік, канал, перетворення і фільтр), які діють на інформаційні потоки. Активний об'єкт відповідає користувачеві в моделі RBAC, а пасивний об'єкт являє собою набір об'єктів в моделі RBAC.

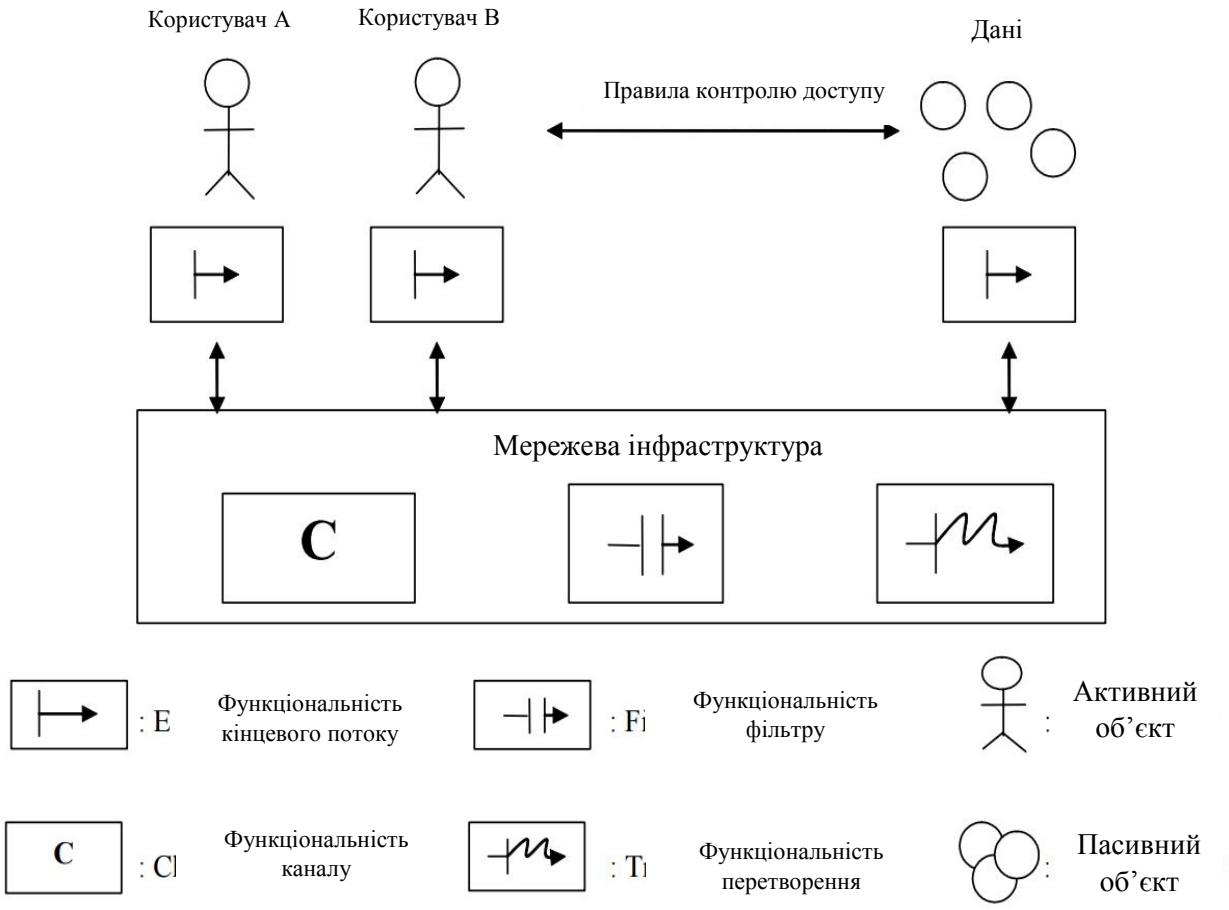

Рисунок 1.3 – Модель мережевої топології

Описується основна ідея моделі з використанням кольорових мереж Петрі (CPN). CPN забезпечують основу для побудови і аналізу розподілених і паралельних систем. Модель CPN системи описує стани, в яких може існувати система, і переходи між цими станами.

1.3.1 Кольорові сітки Петрі

Стани CPN представлені за допомогою місць (які малюються як еліпси або кола). Кожне місце має пов'язаний тип (набір кольорів), що визначає тип даних, які може містити місце. Стан CPN називається маркуванням. Він складається з декількох жетонів, розташованих (розподілених) в окремих місцях. Кожен жетон несе значення (колір), яке відноситься до типу місця, на якому знаходиться. Жетони, присутні на певному місці, називаються маркуванням цього місця. Знаки

CPN відрізняються один від одного, тому вони «пофарбовані», на відміну від мереж Петрі з низьким рівнем, які мають «чорні» нерозрізнені жетони. Маркування місця має безліч наборів значень жетонів. Множинний набір схожий на звичайний набір, за винятком того, що може бути декілька проявів одного і того ж елемента. Це означає, що місце може мати кілька жетонів з однаковим значенням жетона. Наприклад, $1'c_1 + 2'c_2$ означає, що місце містить 3 жетона, одне із значенням c_1 і два зі значенням c_2. Дії CPN представлені за допомогою переходів (які малюються як прямокутники). Переходи і місця пов'язані дугами. Дії CPN складаються з входжень переходів. Виникнення переходу видаляє жетони з місць, пов'язаних з вхідними дугами (місця введення), і додає жетони в місця, пов'язані з вихідними дугами (місця виходу), тим самим змінюючи маркування (стан) CPN. Точна кількість жетонів, доданих і видалених при виникненні переходу, і їх значення даних визначаються виразами дуг. На додаток до виразів дуги до кожного переходу можна приєднати булево вираз (зі змінними). Булево вираз називається охоронцем. Він вказує, що приймаються тільки прив'язки, для яких логічний вираз має значення Правда.

CPN має відмінне маркування – початкове маркування, яке використовується для опису початкового стану системи. CPN може також мати одне або кілька маркувань – мертве маркування, яке не може генерувати будь-яке інше маркування.

CPN не приносять додаткової потужності опису в порівнянні з PN, вони просто дозволяють стисненню інформації. Таким чином, будь-який зазначений CPN може бути пов'язаний з ізоморфним PN. Фаза перетворення CPN в PN називається «розгортання». Згодом, аналіз виконується за допомогою CPN або PN (тимчасова логіка з графом входження, лінійної алгеброю, з матрицею інцидентів, класичними властивостями PN).

1.3.2 Визначення функціональних можливостей моделі CPN

Моделюються потоки даних за допомогою жетонів. Характеристики потоків даних (джерело, пункт призначення, який використовує сервіс) представлені «квітами» жетонів. Кожна функціональність моделюється певною під-мережею CPN, яка діє на жетонах. Потім модель CPN специфікації являє собою взаємозв'язок підмереж. Таким чином, визначається загальна функція, яка відображає всю специфікацію в підмножині CPN. Цей компонентний підхід дозволяє автоматично перетворити специфікацію в еквівалентний CPN.

1.3.2.1 Функціональність кінцевого потоку

Кінцевий потік (EF) - це функціональність, характерна для кінцевих систем, тобто серверів даних і додатків, а також робочих станцій. Він передає потоки додатків/користувачів в мережу. Таким чином, функціональні можливості, які виробляють/споживають потоки даних, визначаються під-набором CPN, який генерує жетони з відповідними кольорами і приймає жетони (рисунок 1.4). Розглядаються два типи функцій кінцевого потоку:

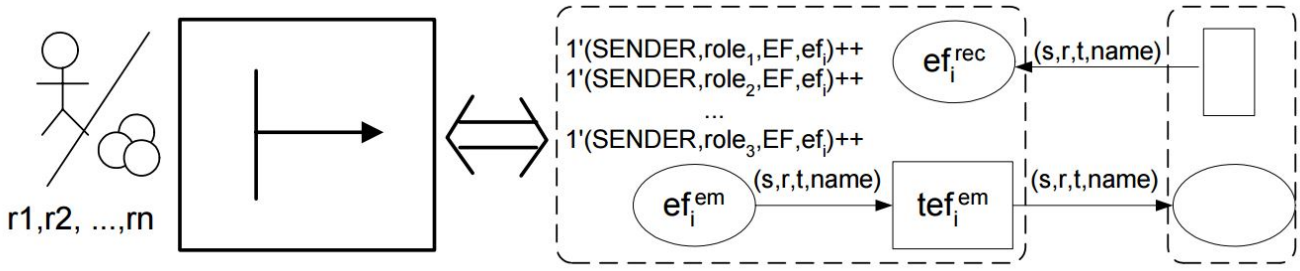

Рисунок 1.4 – Модель CPN з кінцевим потоком

• Функція активного кінцевого потоку (AEF): EF вважається активним, якщо до цього EF підключений будь-якої активний об'єкт.

• Функція пасивного кінцевого потоку (PEF): EF вважається пасивним, якщо до цього EF підключений будь-якої пасивний об'єкт.

Додається список ролей для кожного EF для вказівки потоків, які він може зробити. Список відповідає набору ролей, призначених користувачеві, який представляє підключений активний об'єкт для AEF. У разі PEF це набір ролей, призначених дозволами, які відносяться до пасивного об'єкта.

1.3.2.2 Функціональність каналу

Функціональність каналу моделює фізичну мережу. Вона отримує потік на інтерфейсі і повторно передає його всім пов'язаним об'єктам. Ця функціональність може розглядатися як широкомовний канал. Коли потік орієнтований, він не тільки приймається адресованими адресами, а й усіма системами, підключеними до цього каналу. Функції, які поширюють потоки даних, визначаються під-набором CPN, які отримують жетон з функціональності і відправляють репліку до всіх інших пов'язаних функцій.

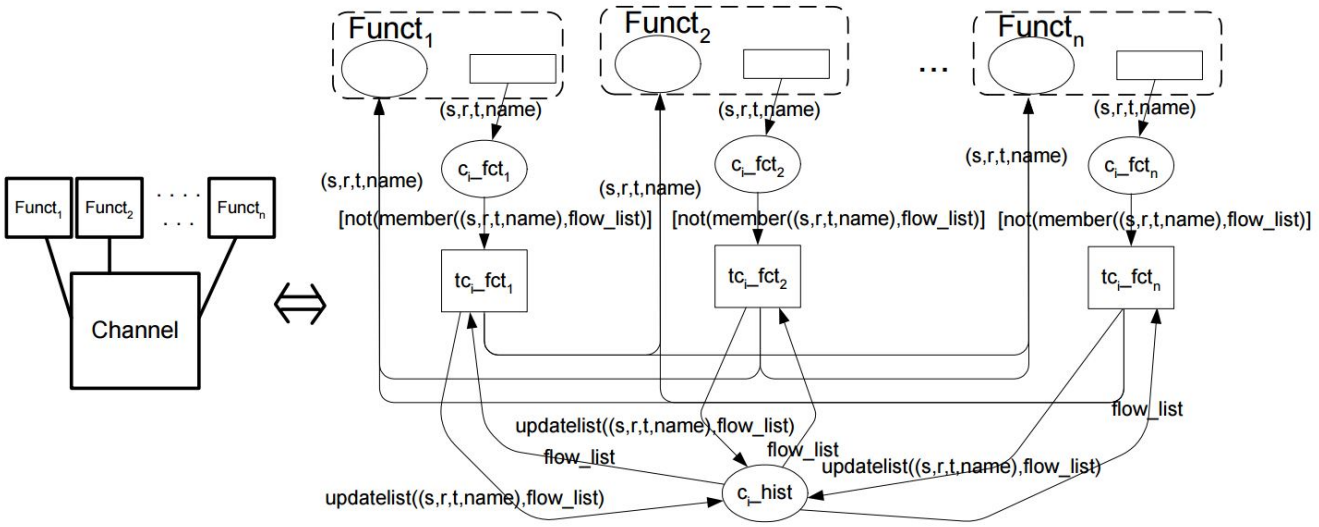

Рисунок 1.5 – Модель CPN для каналу

Таким чином, канальні під-сеанси складаються з набору пар (місце, перехід) для кожної пов'язаної функціональності (рисунок 1.5). Переходи пов'язані з усіма іншими функціональними можливостями. Наприклад, $tc_i\_fct_1$ пов'язаний з $funct_2$, $funct_3$, $funct_n$. Ми також додаємо місце (c_i_hist), яке містить список всіх жетонів, які

пройшли через канал. Він пов'язаний з кожним переходом, щоб гарантувати, що жетон може проходити один раз і тільки один раз через функціональність каналу.

1.3.2.3 Функціональність перетворювання

Функціональність перетворювання отримує потік даних на одному з двох інтерфейсів, і, відповідно до правил перетворення, вона передає через інший інтерфейс цей потік даних або його перетворення.

Трансформація CPN-підмереж змінює колір деяких жетонів відповідно до правил перетворення. Встановлюється функція на переходах після дуг, щоб змінити колір жетона (функція $transf\_funct_1$ і $transf\_funct_2$ на рисунку 1.6). Більш того, якщо функціональність може трансформувати потік, вона повинна мати можливість відновити вихідний потік. Додається два місця ($hist\_tf_i\_fct_1\_fct_2$ і $hist\_tf_i\_fct_2\_fct_1$) для збереження слідів всіх потоків, які пройшли цю функцію.

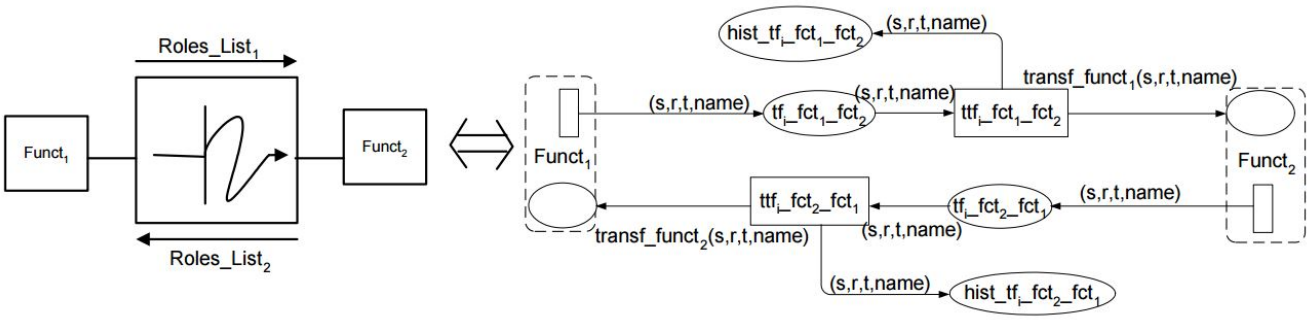

Рисунок 1.6 – Модель CPN з функцією перетворення

1.3.2.4 Функціональність фільтру

Функціональність фільтру зупиняє або пересилає потік даних. Потрібно знайти цю функціональність в брандмауерах, шлюзах рівня додатків або фільтрації маршрутизаторів. Але необхідно її обмежети тільки підключенням двох функцій. Правила фільтрації виражають дозволені потоки між двома його інтерфейсами. Якщо перед ними «EF», то вони надходять неперекладеними з функції кінцевого

потоку, з іншого боку, якщо перед ними «TR», то вони були змінені за допомогою функції перетворення.

Підсистеми CPN фільтра зупиняють або пропускають жетони відповідно до їх кольору і правилу фільтрації (рисунок 1.7). Представляються правила фільтрації, обмежуючи кольори, дозволені переходами з захистом. Тоді не можуть бути запущені жетони з кольором, який не перебуває у захисті переходу. Перехід $tf_i\_fct_1\_fct_2$ (відповідно $tf_i\_fct_2\_fct_1$) використовується для фільтрації потоків даних, що надходять з $funct_1$ (відповідно $funct_2$) в $funct_2$ (відповідно $funct_1$). Крім того, ми додаємо два місця ($hist\_f_i\_fct_1\_fct_2$ і $hist\_f_i\_fct_2\_fct_1$), щоб зберегти всі потоки, які пройшли через цю функціональність.

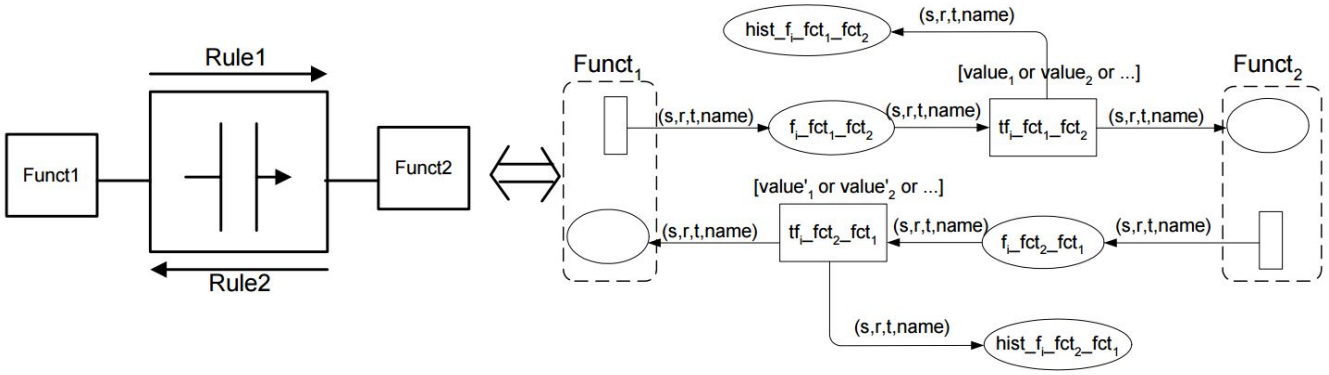

Рисунок 1.7 – Модель CPN функціональності фільтра

1.4 Приклад оцінки політики мережевої безпеки

У цьому прикладі розглядається традиційний випадок мережевої інфраструктури підприємства. Він складається з приватної мережі та DMZ (сегмент мережі, що містить загальнодоступні сервіси і, що відокремлює їх від приватних). Всі з'єднання пов'язані з матеріалом кромки маршрутизатора. У приватній мережі встановлений сервер додатків, а FTP-сервер – в DMZ (рисунок 1.8). Політика безпеки на рівні додатків – це не ієрархічна політика RBAC. Вона визначає дві групи користувачів: групу VPNmembers і групу Others. Ця організація заснована тільки на надані привілеїв. Сервер додатків призначений тільки для служб, які

використовуються групою VPNmembers. Сервер FTP має дві директорії: конфіденційна і публічна. Конфіденційний каталог містить дані, доступні тільки групі користувачів VPNmembers. Дані каталогу «pub» доступні для всіх. Користувач1, Користувач2, Користувач3 і Користувач4 належать членам VPN і іншим користувачам груп. Користувач5 є тільки членом групи «Інші»(«Others»).

На рисунку 1.8 показана специфікація мережевий топології і політика безпеки мережевого рівня. Також додається ім'я, яке використовується в специфікації CPN для кожної функціональності. Цей підхід не враховує пристрої як сутності, а ґрунтується на методах обробки даних. Приватна мережа, DMZ і інфраструктура між-мережевої взаємодії Інтернету, визначаються завдяки функціям каналу, тому що використовуються їх функції передачі. Такий підхід специфікації з великим ступенем деталізації враховує мінімальний набір функціональних можливостей, що надаються цими інфраструктурами: їх здатність до між-з’єднання.

Можна уточнити специфікацію, яка показує її крайовий маршрутизатор. Це функціональність з'єднання (функціональність каналу), налаштування в якості шлюзу безпеки з можливостями фільтрації (три функції фільтра) і механізми шифрування. Моделювання маршрутизації виконується за допомогою правил фільтрації. Сервери задаються двома PEF. Сервер додатків має роль VPN, оскільки тільки користувачі з роллю VPNmembers мають права доступу. PEF, відповідний до FTP-сервера, має ролі Others і VPNmembers, оскільки дозвіл (+ all access, FTP Server / pub) призначається ролі Others (+ all access, FTP Server / конфіденційний) і ролі VPNmembers. Пристрої User1 і User2 представлені одним AEF (EF1), оскільки User1 і User2 мають однакові ролі (Others і VPNmembers), а ці AEF підключені до однієї і тієї ж функції каналу, завдяки концепції ролі, яка зменшує загальний розмір специфікація. Точно так, як користувачам User3 і User4 задаються AEF EF5. Пристрій User5 задається AEF EF4. Довільне додавання AEF з ролями, вирішення яких зменшено, дозволяє визначити ступінь достовірності, щоб отримати повне

визначення «властивості довіри каналу», яке може бути надано функціональності каналу.

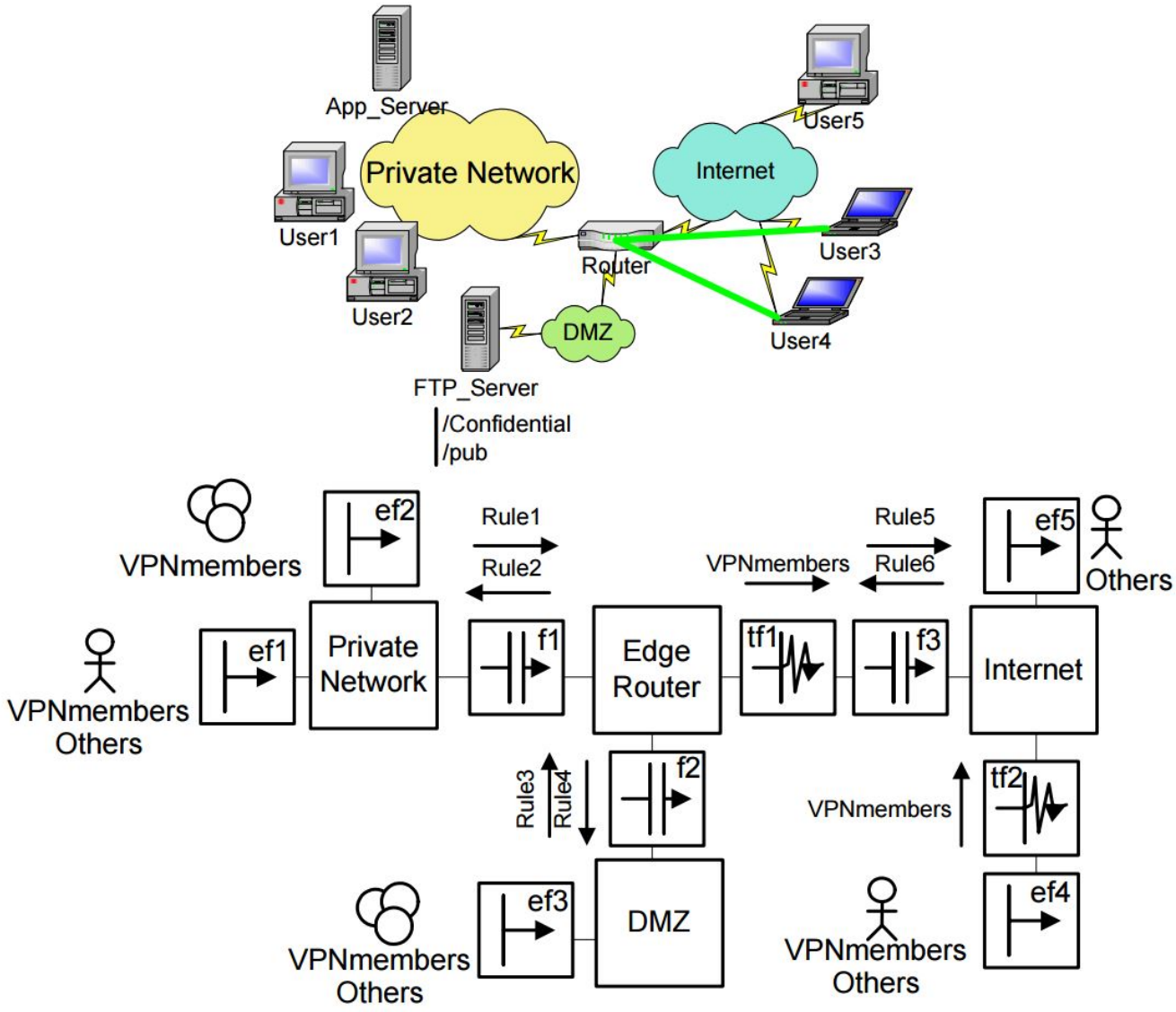

Рисунок 1.8 – Архітектура та специфікація нашого прикладу VPN

Правила фільтрації, пов'язані з функціональними можливостями фільтра:

• Правило 1 = EF (AEF, Others), (AEF, VPNmembers);

• Правило 2 = EF (PEF, Others);

• Правило 3 = EF (PEF, Others), (P EF, VPNmembers), (AEF, VPNmembers);

• Правило 4 = EF (AEF, Others), (AEF, VPNmembers);

• Правило 5 = EF (PEF, Others), (AEF, Others), TR (PEF, VPNmembers);

• Правило 6 = EF (AEF, Others), TR (AEF, VPNmembers).

Визначено дві функції перетворення для захисту потоків даних ролей VPNmembers в функціональності інтернет-каналу. Користувачі, підключені до функцій інтернет-каналу з роллю «Others», ніколи не можуть отримати доступ до конфіденційних даних на FTP-сервері або сервері додатків. Використовується CPN/tool для створення CPN (рисунок 1.10), пов'язаного зі специфікацією. Він показує початкове маркування. На рисунку 1.10 видно, що CPN складний для ручної побудови специфікацій великого розміру. Тому його розроблено, з використанням мови програмування Java, інструмент, який автоматизує задачу оцінки. В якості вхідного файлу потрібен файл специфікації (рисунок 1.9). По-перше, він аналізує синтаксис. Якщо синтаксис вірний, він генерує еквівалентний CPN і перевіряє всі властивості. В результаті виходить файл (рисунок 1.11), який вказує, виконані властивості чи ні.

У цьому прикладі інструмент показує (рисунок 1.11), що властивість конфіденційності виконано і не існує правила непродуктивного перетворення. Проте, доступність не виконується, тому що ef_2 не може отримувати потік з роллю VPNmembers з ef_5, ef_1 не може отримувати потік з роллю VPNmembers з ef_3 і ef_5 не може отримувати потік з роллю VPNmembers з ef_2. Властивість розбиття не виконується через правила EF (AEF, Others) з tf_1 в Internet в функціональності фільтра f_3. І, нарешті, правило фільтрації EF (AEF, VPNmembers) від DMZ до крайнього маршрутизатора в функціональності фільтра f_2 не є продуктивним.

```
/* end-flow functionalities definition */
<AEF>
#name = ef1
#roles = others, vpn-members;
#connection = private_network

<PEF>
#name = ef2
#roles = vpn-members;
#connection = private_network

<PEF>
#name = ef3
#roles = others, vpn-members;
#connection = dmz

<AEF>
#name = ef4
#roles = others;
#connection = internet

<AEF>
#name = ef5
#roles = others, vpn-members;
#connection = tf2

/*transform functionalities definition */

<TRANSF>
#name = tf1
#connection1 = edge_router
#connection2 = f3
#rules_1->2 = vpn-members;
#rules_2->1 = NONE;

<TRANSF>
#name = tf2
#connection1 = ef5
#connection2 = internet
#rules_1->2 = vpn-members;
#rules_2->1 = NONE;

/* filter functionalities defintion */
<FILTER>
#name = f1
#connection1 = private_network
#connection2 = edge_router
#rules_1->2 =
EF (AEF,others), (AEF, vpn-members);
TR NONE;
#rules_2->1 =
EF (PEF, others);
TR NONE;

<FILTER>
#name = f2
#connection1 = dmz
#connection2 = edge_router
#rules_1->2 =
EF (PEF,others), (PEF, vpn-members),
(AEF, vpn-members);
TR NONE;
#rules_2->1 =
EF (AEF, others), (AEF, vpnmembers);
TR NONE;

<FILTER>
#name = f3
#connection1 = tf1
#connection2 = internet
#rules_1->2 =
EF (PEF,others), (AEF, others);
TR (PEF, vpn-members);
#rules_2->1 =
EF (AEF, others);
TR (AEF, vpn-members);

/* channel functionalities definition */
<CHANNEL>
#name = private_network
#connection = ef1, ef2, f1;

<CHANNEL>
#name = edge_router
#connection = f1, f2, tf1;

<CHANNEL>
#name = internet
#connection = f3, ef4, tf2;

<CHANNEL>
#name = dmz
#connection = ef3, f2;
```

Рисунок 1.9 – Файл специфікації

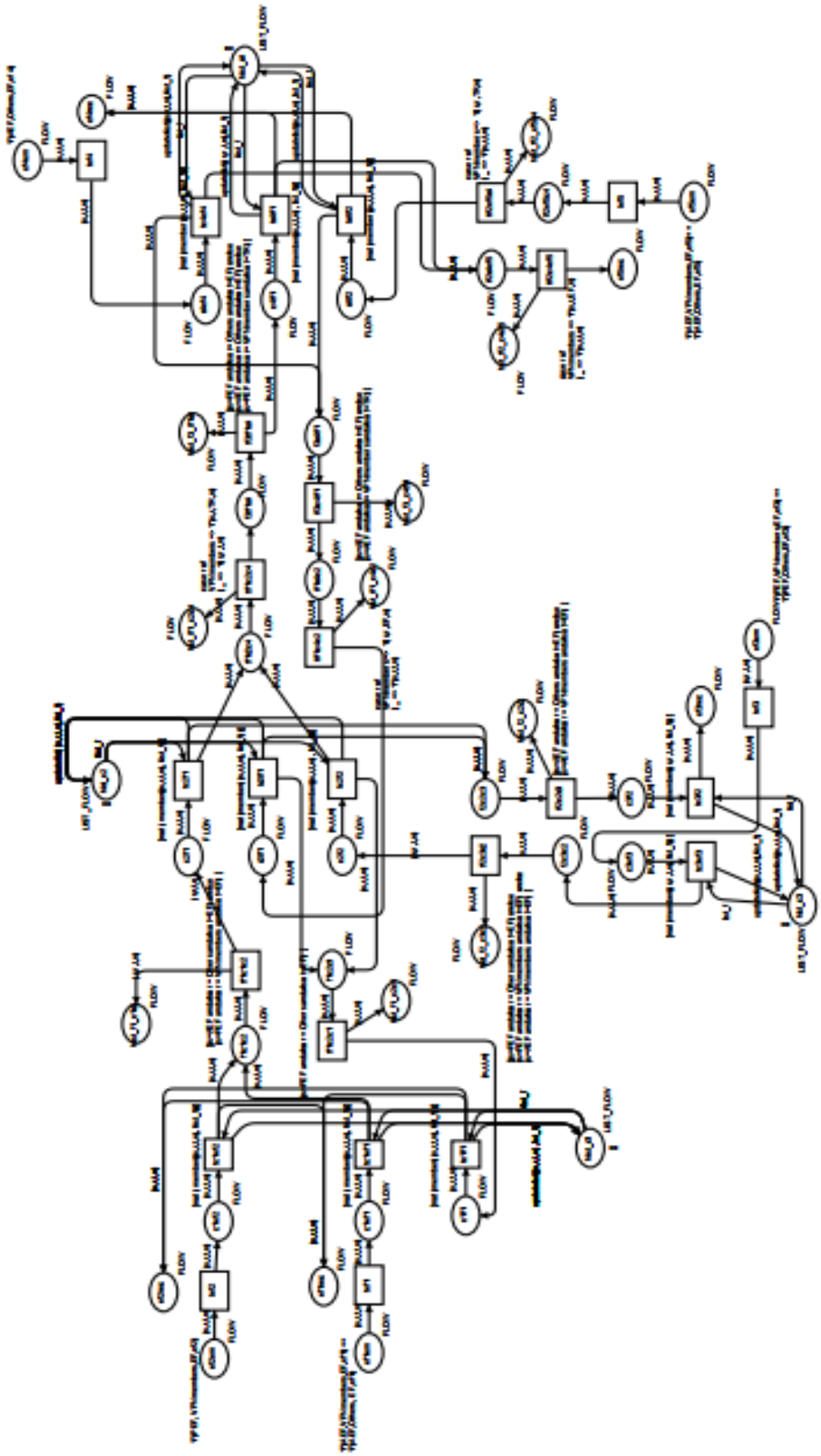

Рисунок 1.10 – Модель CPN прикладу VPN

```
Property of Confidentiality :
-----------------------------
ef5 : OK

ef4 : OK

ef1 : OK

=> The property of confidentiality is satisfied


Property of Availability :
--------------------------
ef5 :
    no flow with the role vpn-members from ef2

ef4 : OK

ef1 :
    no flow with the role vpn-members from ef3

ef3 : OK

ef2 :
    no flow with the role vpn-members from ef5

=> The property of availability is not satisfied


Non Productive Transform Rules :
--------------------------------
tf2 :
    rules 1 -> 2 : OK
    rules 2 -> 1 : OK
tf1 :
    rules 1 -> 2 : OK
    rules 2 -> 1 : OK

=> There is no non productive rule


Non Productive Filtering Rules :
--------------------------------
f3
    rules 1 -> 2 :  OK

    rules 2 -> 1 :  OK

f2
    rules 1 -> 2 : [ EF (AEF, vpn-members) ],
    rules 2 -> 1 :  OK

f1
    rules 1 -> 2 :  OK

    rules 2 -> 1 :  OK

=> There is one or more non productive rule


Partitioning Property :
-----------------------
f3 :
    Rule 1 -> 2 :
       [ EF (AEF ,others) ]
    Rule 2 -> 1 : OK
f2 :
    Rule 1 -> 2 : OK
    Rule 2 -> 1 : OK
f1 :
    Rule 1 -> 2 : OK
    Rule 2 -> 1 : OK

=> There is one or more partitioning problem
```

Рисунок 1.11 – Файл результату оцінки

1.5 Висновки

Дизайн політики безпеки стає дедалі складнішим через складність факторів для розгляду. Загальний підхід до визначення різних рівнів абстракції, від цілей до конфігурацій пристроїв, використовується для подолання проблеми. Деякі існуючі інструменти реалізують цей підхід. Тим не менше, формальна та автоматична оцінка рішень повинна завершити це досягнення.

РОЗДІЛ 2

ОЦІНКИ МЕХАНІЗМІВ МЕРЕЖЕВОЇ БЕЗПЕКИ НА ОСНОВІ ПОЛІТИКИ RBAC

2.1 Моделі RBAC

Sandhu et al. визначив набір моделей RBAC. Пізніше Ферреіоло та ін. пропонує стандарт NIST для RBAC. Різні архітектури для послуг RBAC в Інтернеті пропонуються в Park et al. Проте в цих моделях RBAC використовується статичне асоціація користувальницьких ролей, яка дозволяє налаштувати кожну призначену роль користувача. Крім того, в управлінській моделі користувача, відому під назвою URA97 ARBAC97, необхідні декілька етапів, щоб призначити користувача ролі, тому що умови обов'язкового розподілу користувацьких функцій у URA97 визначаються з регулярними ролями, які утворюють ієрархію ролей. Як наслідок, важко адмініструвати політику, коли модель розгортається у великій корпоративній системі, оскільки вона має багато груп або підрозділів.

ARBAC02 присвячена кількома завданням користувача та завданням дублювання інформаційних завдань, визначаючи відносини з користувачами на основі існуючої структури інформації як пулу користувачів і пулу дозволів, наприклад, позицію користувача від відділу кадрів та дозволи від IT-відділу, замість звичайних ролей.

Обмеженням цієї моделі є те, що ролі в пулі користувачів повинні мати відношення часткового порядку. Це призводить до серйозних обмежень на присвоєння користувача в ситуації, коли існує безліч різноманітних дискретних ролей. Інша слабкість в ARBAC02 полягає в тому, що вона вимагає попередньо визначених пулів користувачів (OS-U) і пулів дозволів (OS-P), і це представляє складні завдання для адміністраторів.

Nyanchama та Osborn пропонують модель графічного графіка, яка еквівалентна ієрархії ролей в моделі RGBC Sandhu. Замість того, щоб визначати відносини

явного розподілу користувальницьких ролей, ця модель також вводить поняття групи для надання неявного розподілу користувацької ролі. Фактично завдання групової ролі дуже схоже на призначення користувальницької ролі в оригінальному RBAC. Таким чином, ця модель не розглядає великі адміністративні завдання для адміністраторів і не повністю використовує компоненти груп для полегшення адміністративної моделі. Крім того, адміністрація моделі ролевого графіку централізована, яка пропонує менш гнучкий підхід.

Проте ці моделі сильно залежать від ієрархії ролі та зосереджуються на модифікації графів; тобто, вони не вважають головною метою для користувача виконувати завдання.

В цілому, модель ARBAC97 зосереджується на централізованій адміністративній моделі, де один спеціаліст із супер безпеки (SSO) визначає ролі та виконує завдання, призначені для користувача та ролі. ARBAC02 використовує структури організації на підприємстві для спрощення цих завдань. Обидві моделі не вирішують проблему в децентралізованому середовищі, де потрібне гнучке та автономне керування авторизацією.

Декілька років тому дослідницькі зусилля були присвячені темі взаємодії в багато-доменних середовищах. В Kapadia запропоновано модель трансляції динамічної ролі та кілька питань безпеки. Шафік, Піромруен та Джоші пропонує ряд безпечних схем взаємодії. У Джоші запропоновано RBAC на базі XML для визначення багато-доменних політик.

Пропонуються рішення, які базуються на моделі Generalized Temporal Role-Based Access Control (GTRBAC). Проаналізовано три типи порушень при інтеграції політики RBAC: порушення користувачем роздільного виконання службових обов'язків, порушення роле-специфічного порушення правил і порушень ролевих завдань. Наприклад, порушення права доступу до ролі відбувається, коли користувачеві домену дозволяється отримати доступ до ролі, навіть якщо користувач не призначений безпосередньо для ролі чи будь-якої з ролей, старших

за роль в ієрархії ролей у домені . Крім того, Piromruen et al. перетворило локальну політику GTRBAC для полегшення між-доменних операцій. Ці підходи використовують підхід «з висхідним підйомом» до складеної політики RBAC, і їм доводиться вирішувати багато проблем при виникненні політик, таких як роль прихованої реклами та всіх видів вищезазначених порушень. Толоне та ін. обговорюють вимоги до контролю доступу в спільних системах та аналізують існуючі моделі доступу, включаючи RBAC в спільних середовищах.

2.2 Недоліки традиційних моделей RBAC

Традиційні моделі RBAC, такі як RBAC96 та стандарт NIST RBAC, не можуть забезпечити ефективне управління авторизацією для співпраці. Основна причина полягає в тому, що ці моделі зосереджують увагу на контролі прав користувачів у відповідності з попередньо встановленими ролями та відносинами для розподілу повноважень. Перебуваючи в динамічних середовищах, взаємодія ролей та призначень для користувача-ролі не фіксується під час співпраці. З іншого боку, оскільки кількість ролей та користувачів у системі RBAC коливається від десятків до тисяч у великих корпоративних системах, управління авторизацією є великою перешкодою для безпечного доступу. Багато адміністративних моделей для RBAC було запропоновано для зручності та ефективності управління. Проте більшість (якщо не всі) цих моделей визначають політику адміністрування, засновану на існуючих ієрархіях ролей, такими, що вони не можуть підтримувати динамічне та одночасне співробітництво між групами, оскільки ієрархія ролей у цих сценаріях не є статичною.

Крім того, попередні дослідження на RBAC вважали обмежену дію дозволу, такі як читання, запис та виконання в операційній системі та вставка, видалення, виділення та оновлення в базі даних. Однак у випадку, якщо робота дозволу є виконуваним кодом, як програмні компоненти, використання яких останнім часом стрімко зростає, функціонування дозволу відображається у вигляді різних операцій

і взаємодіє з іншими операціями за допомогою виклику. У цьому випадку, оскільки між операціями існує зв'язок залежності та асоціацій, необхідно відобразити ці відносини на обмеження та призначення дозволу.

Проблема симетричних моделей RBAC полягає в тому, як призначити лише відповідні дозволи для ролі під час інтеграції дозволів через ієрархії ролей або агрегування боргу та функцій тощо. Взагалі, старша роль успадковує дозволи своїх молодших за ієрархією. У цьому процесі, якщо дозвіл неприйнятний для ролі, то вона виконує діяльність, що перевищує те, що було дозволено. Отже, слід представити симетричну модель RBAC, здатну запобігти неналежне успадкування дозволів.

2.3 Альтернатива традиційним моделям RBAC

Вирішення проблеми полягає в спрощенні децентралізації адміністративних завдань і, таким чином, підвищення практичності використання RBAC у динамічних середовищах співпраці. У даній роботі ми пропонуємо децентралізовану та групову модель RBAC (GBRBAC), ввівши поняття груп та модифікацію моделі призначення користувацької ролі.

Модель GB-RBAC зберігає основні риси RBAC і спрощує призначення завдань користувача за допомогою двох механізмів. По-перше, GB-RBAC надає стандартний набір групових ролей (DSet) для зменшення адміністративних завдань через компонент групи. Таким чином, новий член групи може бути призначений з набором ролей за замовчуванням без участі адміністратора. По-друге, адміністратор групи може призначати інші чіткі ролі учаснику групи, керуючись локальною адміністративною політикою, наскрізним чином. У цій моделі впроваджено спрощений, але гнучкий розподіл користувацьких функцій. Тому наша модель забезпечує дворівневу адміністративну модель, що полегшує адміністративні питання RBAC на глобальному або системному рівнях та на місцевому або груповому рівнях. Таким чином, можна досягти таких переваг:

- Модель, природно, підтримує децентралізоване управління простим та ефективним способом. Наприклад, в системах групового співробітництва адміністратори груп можуть додавати або змінювати завдання для задоволення вимог програми та місцевої адміністрації без участі глобальних адміністраторів. Це спрощує системні рівні адміністративних завдань та забезпечує гнучкий механізм адміністрування динамічних призначень для користувачів, особливо в спеціальних середовищах співпраці.

- Адміністративна модель забезпечує налаштовуванні адміністративні дозволи групового рівня, які контролюються адміністраторами системного рівня. Таким чином, значно спрощується не тільки призначення користувальницького завдання для системного адміністратора, але також реалізується принцип розподілу боргу (SD) на адміністративному рівні.

- Адміністративна модель задовольняє вимогам адміністрації автономії для RBAC, такими як дрібнотоварні розподіли користувачів та налаштовувана адміністрація групового рівня.

Реалізація моделі:

- Обирається модель GB-RBAC, яка базується на моделі RBAC96 і розширена за допомогою групової концепції. Ми визначаємо різницю між нашою моделлю та іншими груповими підходами.

- Розробляється дворівневу адміністративну модель для GB-RBAC з функціями децентралізованого управління, налаштовуваними адміністративними дозволами групового рівня та принципом адміністративного режиму.

- Розробляється механізм використання нашої моделі для безпечного одночасного співробітництва між групами шляхом введення поняття віртуальної групи. Ми визначаємо алгоритми для виконання завдань користувача та ролі у віртуальних групах.

- Розробляється служба авторизації на основі моделі GBRBAC, а прототип моделі показує доцільність у реальних розподілених додатках. Наша модель забезпечує прийнятну продуктивність для спеціальних додатків співпраці.

2.4 Модель GB-RBAC

2.4.1. Огляд GB-RBAC

В даній роботі запропонована модель GB-RBAC вводить концепцію групи, за допомогою якої створюються нові механізми розподілу користувальницьких функцій. Суттєва різниця між групами та ролями полягає в тому, що група – це сукупність користувачів, які мають аналогічні атрибути безпеки, а роль – сукупність дозволів.

На рисунку 2.1 показані всі компоненти в моделі GB-RBAC та як це працює. Користувачі, ролі, ресурси, операції та дозволи подібні до моделей RBAC96. Як правило, дозволи асоціюють операції з ресурсами, а права і користувачі призначаються для ролей. GB-RBAC представляє два нових компоненти: групи та лідери групи (адміністратори). Група, як згадувалося вище, являє собою набір користувачів, а адміністратор групи – це користувач, призначений для ролі адміністративної групи. Крім того, призначаючи користувачам ролі системними адміністраторами, GB-RBAC підтримує призначення користувачами роль адміністраторами групи.

На рисунку 2.1 є два типи рядків: суцільні лінії позначають зв'язки між компонентами та штриховими лініями позначають керування цими асоціаціями. Через нову складову груп ми можемо легко визначити, що у попередніх моделях пунктирні лінії серед груп, ролей, дозволів і операцій не відображаються. Таким чином, GBRBAC може надавати два рівні адміністрування користувачів:

- Перша – це адміністрування на рівні системи, пов'язане з централізованим контролем над призначенням користувача, тобто означає пунктирні лінії, початкова точка яких є вузлом системного адміністратора.

- Друга – адміністрація на рівні групи, пов'язана з децентралізованим контролем над призначенням для користувача, яка позначається пунктирними лініями, початковою точкою яких є вузол лідерів групи.

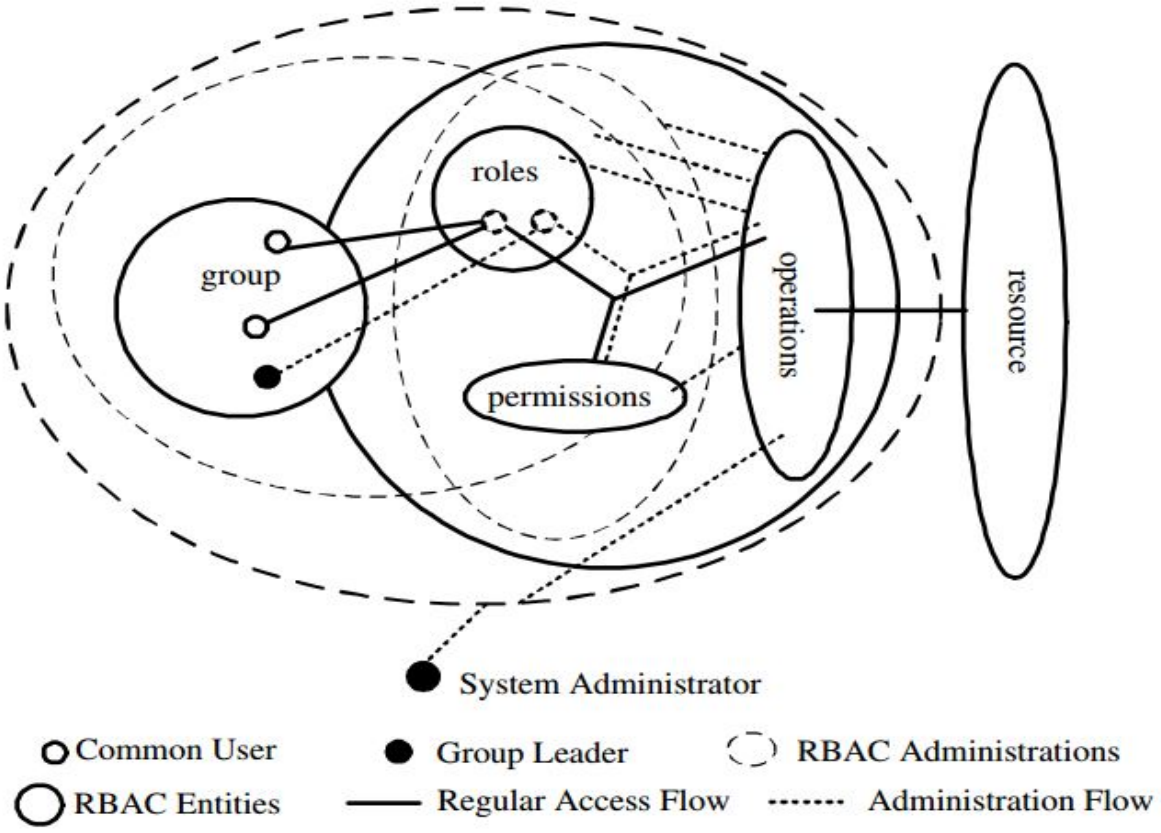

Рисунок 2.1 – Огляд системи GB-RBAC

2.4.2. Опис моделі GB-RBAC

GB-RBAC включає компонент груп в модель RBAC96 і забезпечує децентралізоване управління роллю. GB-RBAC побічно накладає контроль доступу на дію користувача після того, як цей користувач автентифікується і призначений для набору ролей за замовчуванням. На рисунку 2.2 показані компоненти моделі GB-RBAC. Поняття користувачів (U), ролей (R), ієрархії ролей (RH), дозволів (P), призначення розподілу повноважень (PA) та сеансів (S) ідентичні оригінальній моделі RBAC96. Крім того, GB-RBAC містить набір груп (G). Кожна група призначається набором ролей («assignment role» - «role role» або GA).

Користувач може належати до однієї чи кількох груп, який відображається у вигляді групування користувачів (UM). Крім того, ми пропонуємо два шари ролей, які називаються ролями системного рівня (SR) та ролями групового рівня (GR). Ролеві ієрархічні відносини (RH) входять до цих двох шарів ролей, подібних до RBAC96. Як показує рисунок 2.2, RH існує в обох ролях рівня в GB-RBAC.

Крім розподілу користувальницьких функцій у системній області, яка схожа на призначене для користувача роль в URA97, існує ще один тип призначення для користувача-ролі, що відбувається в групі. Зокрема, оскільки UM пов'язує користувачів з групами, а GA пов'язує ролі з групами, адміністратор групи може призначити користувача в модулі уніфікованого обміну повідомленнями роль у GA, яка іменується назвою групової функції для користувача (GUA), тоді як оригінальну назву, призначену для користувача на роль системи, на рівні системи (SUA). Іншим словом, GUA служить механізмом, через який роль може бути призначена користувачеві, оскільки користувач має відношення відображення з групою, а роль призначена для групи; а потім користувач має дозволи на доступ до ресурсів, визначених у роль. Як видно з рисунку 2.2, GUA побудовано за взаємозв'язками UM та GA. Тобто, GUA ефективний лише тоді, коли користувачі перебувають у контексті груп, призначених ролями. Через механізм відображення користувачів до груп (UM) та призначення ролі групам (GA) в GB-RBAC вводиться нова концепція групових ролей за замовчуванням (DSet), яка вказує на набір ролей, призначених користувачеві в групі за замовчуванням.

2.4.3 Формальне визначення окремих компонентів в GB-RBAC:

Визначення 1. Модель GB-RBAC має наступні компоненти:

- U, P, SR, GR, S та G (користувачі, дозволи, ролі на рівні системи, ролі на рівні групи, сеанси та групи відповідно);

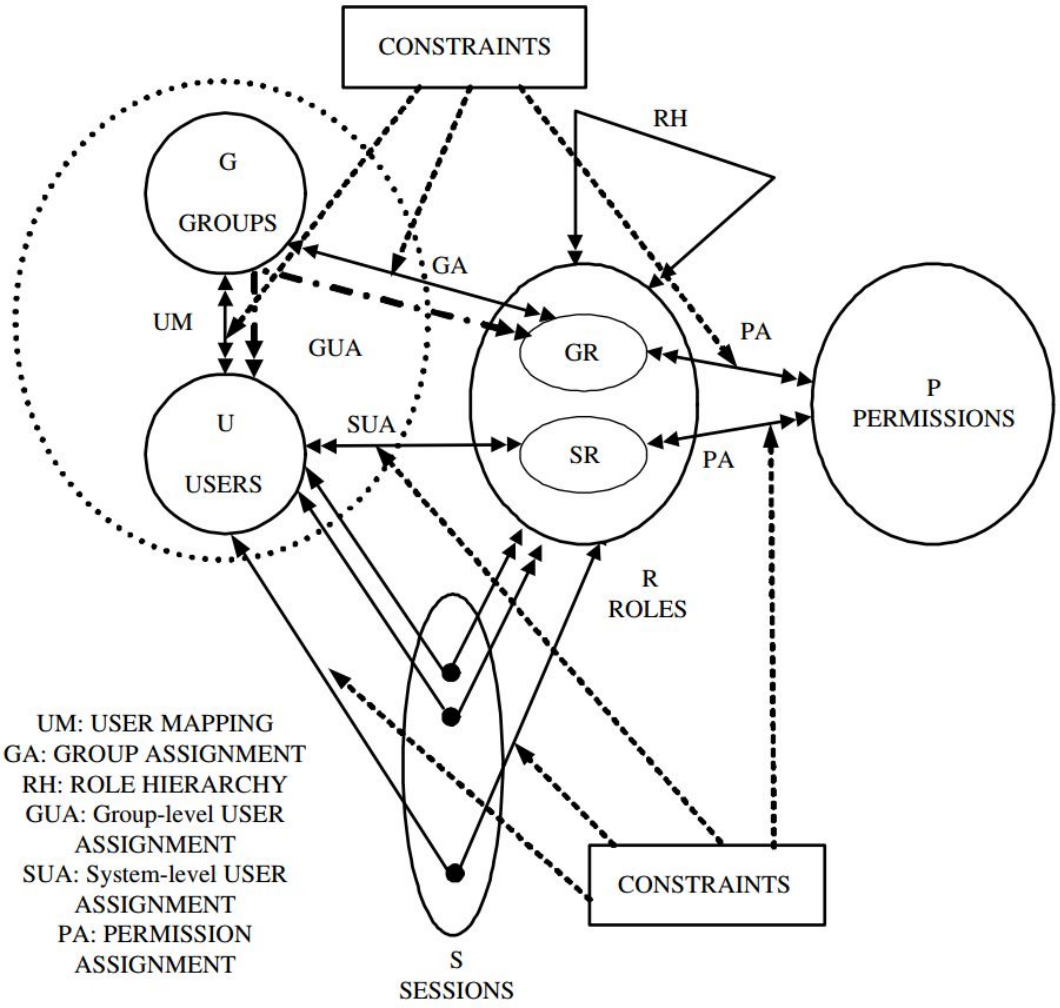

Рисунок 2.2 – Модель GB-RBAC.

- $R = SR \cup GR$. Для простоти, ми беремо $SR \cap GR = ø$ в даному документі;

- $PA \subseteq P \times R$ - дозвіл багатокористувацького відношення до ролевих завдань;

- $UM \subseteq U \times G$, багатокористувацький користувач для групування зв'язку. Це співвідношення показує, що користувач може бути перетворений на різні групи;

- $GA \subseteq G \times R$, група "багатокористувацька" до ролевого завдання;

- $SUA \subseteq U \times SR$, завдання на рівні користувача на рівні системи;

- $GUA \subseteq U \times GR$, призначене для користувача на рівні групи та $(u, r) \in GUA$, тільки якщо $((u, g) \in UM) \wedge ((g, r) \in GA)$;

- UA = SUA ∪ GUA, співвідношення між різними призначеними для користувача рольові завданнями;

- RH ⊆ R × R, часткове замовлення на R називається ієрархією ролі або ролі домінантного відношення. Для будь-яких двох ролей r_1 та r_2, $r_1 \geq r_2$ означає, що r_1 має часткове відношення над r_2;

- користувач: S → U, функція, яка відслідковує кожний сеанс s для одного користувача. Користувач(s) є постійним протягом s;

- дозволи: R → 2^P – функція, яка відбиває роль набору призначених дозволів.

- ролі: S→2^R – функція, що відбиває сеанс на набір ролей, та ролі(s) {r|(∃ r’ ≥ r) [(користувач(s), r’) ∈ UA]}, який може змінюватися протягом сеансів s , а сеанси s мають права доступу $U_{r \in roles(s)}${p|(∃ r’ ’ ≤r) [(p, r’ ’) ∈ PA]}.

Через концепцію групи в визначенні 1 вводяться поняття заданого групового набору ролей (DSet).

Визначення 2. Набір ролей за замовчуванням для групи DSet: G→2^R - це підмножина R, і ∀u ∈ U, r ∈ R, (u, g) ∈ UM ∧ r ∈ DSet (g) → (u, r) ∈ GUA.

За допомогою цього підходу користувач, який підключений до групи, автоматично отримує всі ролі в DSet групі. Процедура визначення дозволів користувача в GB-RBAC описується наступним чином. Коли користувач вводить систему або запускає програму, створюється сеанс і під-набір призначених ролей користувача активується. Набір призначених ролей користувача включає безпосередньо призначені ролі користувача (через SUA), ролі в DSet групи, зареєстровані користувачем, і ролі, призначені адміністраторами групового рівня (через GUA). Користувач отримує всі дозволи, призначені для цих ролей через PA. Користувачі також можуть змінювати активовані ролі в сеансі в рамках його призначених ролей. Сеанс може бути завершений користувачем або системою, наприклад, через тривалість простою в режимі очікування. Для простоти, за один сеанс користувач не може змінити членство в групі.

У GB-RBAC пропонуються два рівні ролей через групи: ролеві рівні системи (SR), які працюють у контексті загальної ролі системи та групи на рівні (GR), які працюють в контексті груп. Таким чином, крім SR, користувач може бути призначений GR, якщо він/вона належить до деяких груп. Користувач, призначений для SR і GR, отримує різні області дозволів. Крім того, DSet, наданий в GB-RBAC, дозволяє новому механізму присвоювати користувача для зменшення завдання адміністратора. Таким чином, для нового учасника групи можуть бути призначені деякі ролі за замовчуванням без участі адміністраторів, а адміністратори груп можуть призначати інші явні ролі для групування членів за ролями в DSet. На рисунку 2.3 показані ці два механізми розподілу користувацьких функцій. У верхній частині вказано SUA, який прямо призначає користувачам ролі. Нижня частина вказує на GUA, де спочатку користувачі переносяться на групи, а потім призначаються для ролей у відповідних групах.

З офіційного опису моделі ми бачимо, що використання ролей в DSet групи не набуває чинності, коли ролі в наборі не мають доручення. Крім того, DSet або група може бути змінена системними або груповими адміністраторами, тому це впливає на загальну призначену роль користувача в GB-RBAC. У моделі GB-RBAC також можуть бути обмеження, визначені багатьма аспектами, показаними на рисунку 2.2. Крім обмежень на SUA, PA, RH та сеанси, подібні до тих, що знаходяться в RBAC96, GB-RBAC вводить нові обмеження на UM, GA та GUA.

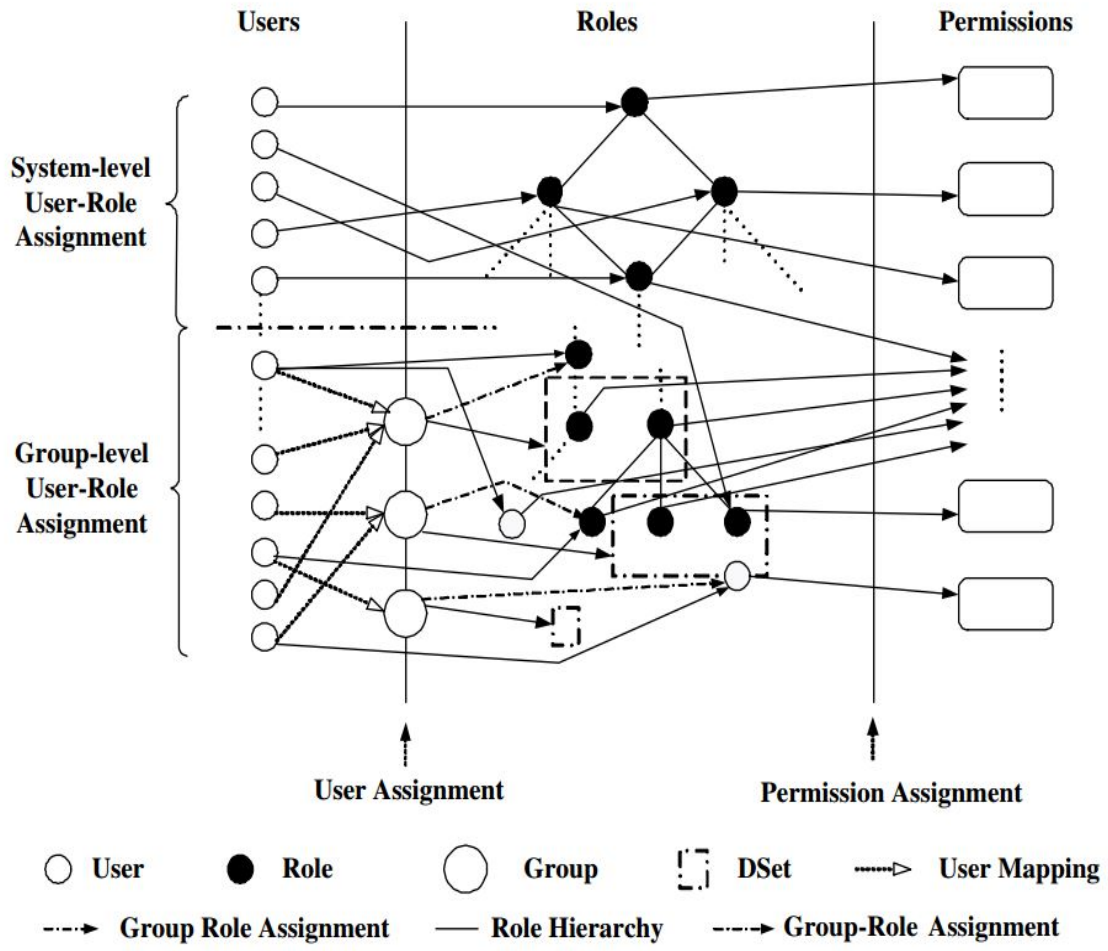

Рисунок 2.3 – Дворівневі призначені для користувача ролі в моделі GB-RBAC.

2.5 Адміністративна модель GB-RBAC

У цьому розділі вперше наведено огляд адміністрування користувача в GB-RBAC, а потім описується формальна модель, а потім обговорюються її переваги над традиційними адміністративними моделями.

2.5.1 Огляд

Успіх системи контролю доступу сильно залежить від його адміністрування, особливо коли число користувачів і ролей становить тисячі. Управління компонентами контролю доступу та їх взаємозв'язків - це важливе та складне

завдання. У порівнянні з попередніми моделями RBAC, наша модель вводить додаткові завдання для адміністрування, такі як призначення групового роду та відображення груп користувачів. Для управління відносинами, визначеними в GB-RBAC, пропонується запропонувати дворівневі адміністративні моделі, що називаються адміністративною моделлю на рівні системи та на рівні групи. Адміністративна модель групування груп користувачів (UM) полягає в тому, щоб класифікувати користувачів до різних груп, і це є обов'язком адміністративної моделі на рівні системи.

Завдання групової ролі (GA) до певної міри схоже на присвоєння ролі користувачеві, яке нараховується адміністраторами системного рівня. Крім того, DSet групи – це ще одне адміністративне завдання, яке може впливати на розповсюдження дозволу в моделі. Управління DSet групи може бути реалізоване на обох рівнях адміністрації. Однак ми вважаємо це адміністрацією групового рівня, оскільки однією з мотивів цієї моделі є надання адміністрації автономності на рівні групи, таке, щоб адміністратор групи мав дозвіл призначати користувачів ролі у співвідношенні з GA.

Для цих двох рівнів адміністрування в адміністративній моделі визначаються два типи адміністративних ролей, які називаються адміністративними ролями системи (SAR) та адміністративними ролями на рівні груп (GAR). Ці адміністративні ролі також можуть утворювати ієрархії ролей, відповідно, аналогічні функції звичайних ролей GB-RBAC. Для простоти, припускається, що $SAR \cap GAR = ø$. Користувач системної адміністративної ролі (або просто системний адміністратор) може призначити користувача групі (через уніфікований обмінний інтерфейс), але користувач ролі адміністратора групи (або просто адміністратор групи) може визначити, на яку роль може користувач бути призначений. Таким чином, забезпечується тип розподілу обов'язків на різних рівнях адміністрування. Крім того, UM може управлятися адміністративною моделлю простішою, ніж GA, і не додає багато складності в адміністративну модель GB-RBAC.

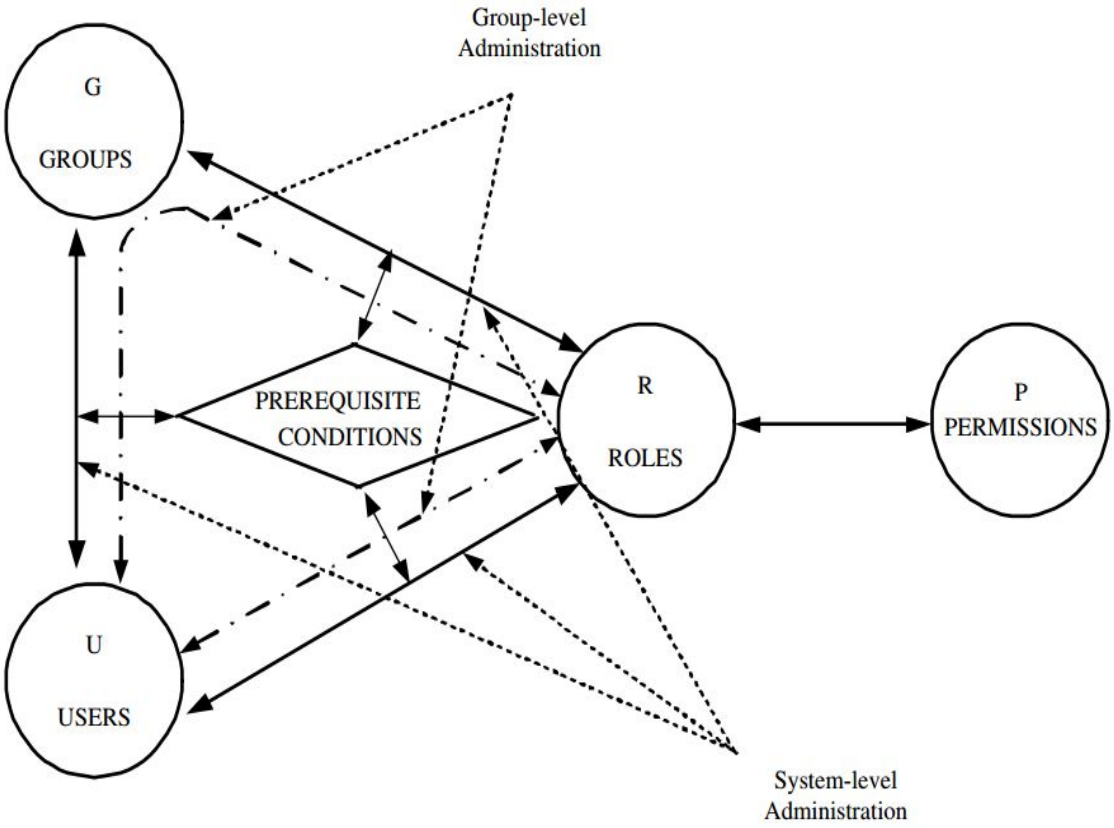

Рисунок 2.4 – Адміністративна модель GB-RBAC

На рисунку 2.4 пропонується дворівнева адміністративна модель, яка включає в себе системні та групові адміністративні рівні, відповідно, для вирішення всіх видів компонентів, визначених у GB-RBAC. Адміністративна модель, розглянута в даному документі, зосереджена на присвоєнні користувальницької ролі, яка стосується управління наступними компонентами в GB-RBAC: GA, UM, SUA, GUA та DSet. Зокрема, адміністративна модель на рівні системи має три типи елементів керування, що застосовуються відповідно до GA, UM, SUA, тоді як адміністративна модель на рівні групи має два типи елементів керування, що застосовуються відповідно до GUA та DSet. Зауважимо, що ми не звертаємось до адміністрації UA, оскільки це може бути здійснено адміністрацією SUA і GUA через визначення 1. Крім того, слід зазначити, що, оскільки контроль над DSet групи визначає типовий набір ролей членів групи, він неявно керує призначення користувача-ролі.

Наприклад, у телефонній конференції на часовій групі роль у спільному дозволі підключення віртуального конференц-залу визначається в DSet, який за замовчуванням призначається всім користувачам у групі. Оскільки група DSet дуже специфічна для застосування, ми не визначаємо явних правил для керування DSet у цьому документі.

Різні рівні адміністрацій забезпечують механізм автономії, такий, що локальні адміністратори групи можуть призначити члену групи різними ролями, якщо деякі умови присвоєння виконані (введено найближчим часом). Інша гнучкість в управлінні UA, крім механізму, наданого DSet, дозволяє вивести адміністративні завдання в адміністративну модель першого рівня.

Адміністративні моделі різного рівня мають різні обов'язки: адміністрування компонентів GB-RBAC з централізованим контролем над користувачами знаходиться в адміністративній моделі на рівні системи, а адміністрування в групі є адміністративною моделлю на рівні групи. На рисунку 2.4 показано обсяг цих дворівневих адміністрацій. Модель системи рівня діє в системі, щоб призначити/відкликати ролі системного рівня до/від користувачів, призначати/відкликати ролі групового рівня до/з груп, а також додавати/вилучати користувача до/з груп, тоді як працює модель групи рівня в групі, щоб призначити/відкликати ролі до/від користувачів, включаючи визначення ролей в DSet. Для простоти не розглядається адміністрування дозволів та ієрархії ролей.

2.5.2. Модель гранту у GB-RBAC

Модель гранту визначає правила, які дозволяють адміністраторам призначати користувачів ролі та групи. Перш ніж пояснити подробиці цих правил, вводяться поняття попередніх умов у різних видах завдань.

2.5.2.1. Користувацькі та групові попередні умови

Визначення 3. Стан умови користувача – це логічне вираження з використанням звичайних операторів $\wedge$ і $\vee$ на умовах форми x та $\overline{x}$, де x - це регулярна роль (тобто $x \in R$) або група (тобто $x \in G$) . Попередня умова оцінюється для користувача u шляхом тлумачення x, щоб бути істинним, якщо будь-яке з наступного є істинним:

- якщо $x \in R$, $\exists x' \geq x$, $(u, x) \in UA$;

- якщо $x \in G$, $(u, x) \in UM$.

І тлумачення $\overline{x}$ буде істинним, якщо будь-яке з наступного є істинним:

- якщо $x \in R$, $\forall x' \geq x$, $(u, x) \notin UA$;

- якщо $x \in G$, $(u, x) \notin UM$.

Для заданого набору ролей R і G, нехай CR_u позначає всі можливі умови, які можуть бути сформовані.

Попередні умови для користувача перевіряють членство або неприєднання користувачів ролей та груп. Членство в ролевих тестах як SUA, так і GUA, умови, що визначаються вище, є виразнішими, ніж у URA97.

Визначення 4. Попередні умови групи є логічним виразом, використовуючи звичайні $\wedge$ і $\vee$ оператори на умовах вигляду x та $\overline{x}$, де x – регулярна роль (тобто $x \in R$). Попередня умова оцінюється для групи g, тлумачивши x, щоб бути правдою, якщо $\exists x' \geq x$ $(g, x') \in GA$, а тлумачення $\overline{x}$ дорівнювати, якщо $\forall x' \geq x$, $(g, x') \in GA$. Для заданого набору ролей R, нехай CR_g позначає всі можливі умови умовної групи, які можуть бути сформовані.

Умови групової передумови перевіряють відношення GA, щоб перевірити членство/нечленство групи, яка використовується для адміністрування призначення групової ролі.

2.5.2.2 Призначення для користувача

Авторизації присвоєння користувачами ролі GB-RBAC контролюються набором правил.

Визначення 5. У системі адміністративного гранту на рівні системи:

- призначення користувацької ролі в SUA контролюється за допомогою співвідношення can_assign_SUA $\subseteq$ SAR $\times$ CR_u $\times$ 2^R.

- картографування груп користувачів в режимі уніфікованого обміну даними контролюється за допомогою співвідношення can_assign_UM $\subseteq$ SAR $\times$ CR_u $\times$ 2^G.

- призначення групової ролі в GA регулюється за допомогою співвідношення can_assign_GA $\subseteq$ SAR $\times$ CR_g $\times$ 2^R.

Визначення 6. У моделі адміністративного гранту на рівні групи:

- призначення завдання користувача в GUA регулюється співвідношенням can_assign_GUA $\subseteq$ GAR $\times$ CR_u $\times$ 2^R.

Співвідношення у двох визначень вище має три параметри (x, y, {z}), що означає, що член x може призначити користувача/групу членом ролі в діапазоні ролей {z}, якщо користувач/група задовольняє умовам відповідної передумови y:

- Відношення can_assign_SUA(x, y, {z}) або can_assign_GUA(x, y, {z}) означає, що член системи або групової адміністративної ролі x (або адміністративна роль, старша до x) може призначити користувача бути членом ролі в діапазоні ролей {z}, якщо користувач задовольняє умовам попереднього умови y.

- Відношення can_assign_UM (x, y, {z}) означає, що член системи адміністративної ролі x (або адміністративна роль, старша до x) може призначити користувача бути членом групи в {z}, якщо користувач задовольняє передумова умови y.

- Відношення can_assign_GA (x, y, {z}) означає, що член системи адміністративної ролі x (або адміністративна роль, старша до x) може призначити групу ролі в діапазоні ролей {z}, якщо група задовольняє передумови умови y.

В моделі GB-RBAC користувачі (наприклад, облікові записи користувачів), ролі та дозволи створюються системними адміністраторами, а адміністратори груп можуть керувати своїми відносинами на рівні групи.

Як показано на нижній частині рисунка 2.3, порожні кола в колонці "roles" позначають адміністративні ролі, а тверді кола в колонці "roles" позначають спільні ролі. Якщо користувач у групі призначений для адміністративної ролі, він/вона може призначити ролі членам групи після успішного тестування обов'язкової умови. Таким чином, члени групи можуть бути належними ролями через GUA. Крім того, верхня частина на рисунку 2.3 ілюструє інший механізм розподілу в користувальницькій ролі, за допомогою якого користувач також може бути призначений ролями через SUA.

Таблиця 2.1 – Правила адміністраційного контролю

Тип	Адміністративна роль	Попередня умова	Діапозон груп/ролей
can_assign_SUA	E-SSO	resAA	resAD
can_assign_UM	E-SSO	resAA	{@PRO1}
can_assign_GUA	PM	@PRO1⋀$\overline{Q}\overline{E}$1	{PE1}

Як приклад, розглянемо набір правил адміністрування, визначених в організації, як показано в таблиці 1. Ми поміщаємо "@" перед назвами групи, щоб відрізнити їх від назв ролей. Група (PRO1) створена для створення адміністративного домену групового рівня. Ролі створюються, як показано на рисунку 2.5. Частина над пунктирною лінією представляє ролі системи на рівні системи, а частина, розташована нижче пунктирної лінії, відображає ролеві рівні на рівні групи. Ці два рівні ролей містять два типи ролей: звичайні ролі, такі як resAA (доступ до ресурсу A) у ролях на рівні системи та PL1 в ролях на рівні групи, а також адміністративні ролі, такі як S-SSO на системному рівні ролі та GD у групі

рівних ролей. Рольові ієрархії також існують серед цих ролей. Наприклад, є ієрархія ролей між двома системними рівнями ролей: наймолодша роль resAA і старша роль resAO (ресурс A власник). Між ними існує ще дві незрівнянно важливі ролі: resAD (розповсюдження ресурсу A) і resAM (модифікація ресурсу A).

Призначення ролі на рівні системи подібне до того, що існує в URA97. Наприклад, якщо Аліса є членом системи E-SSO, а Боб є членом resAA, вона може призначити Боба ролі resAD відповідно до першого правила в таблиці 1. У той же час, згідно з другим правилом, Аліса може призначити Боба групі PRO1. Тепер ми розглядаємо адміністрування на рівні групи. Ми припускаємо, що Керол є членом PM, а Боб є членом групи PRO1. Керол може призначити Боба PE1, якщо Боб не є членом QE1, відповідно до can_assign_GUA(PM, @ PRO1 ∧ $\overline{\text{QE}}$1, {PE1}).

В GB-RBAC, відображення ролей у групах (тобто, щоб визначити GA) дуже специфічне для застосування. Адміністративна модель не включає в себе правила для цілей. Наприклад, на рисунку 2.5 припускаємо, що ролі в PRO1 заздалегідь визначені.

Припущення в адміністративній моделі полягає в тому, що системний адміністратор довіряє не призначати ролі конфліктуючим групам. Наприклад, у вищезгаданому випадку, оскільки рольовий діапазон [ER1, PL1] був призначений групі PRO1, Аліса та інші адміністратори не повинні призначати будь-яку роль в цьому діапазоні іншої групи (скажімо, PRO2), в іншому випадку користувач з PRO2 може мати дозволи в PRO1, наприклад, для читання/запису конфіденційних даних, що взагалі заборонено. Це припущення також використовується в традиційних адміністративних моделях RBAC, наприклад, адміністратори довіряють не призначати дві конфліктні ролі одному користувачеві.

2.5.3. Модель відкликання GB-RBAC

Правила скасування в GB-RBAC регулюються відносинами can_revoke.

Визначення 7. У системі адміністративного відкликання на рівні системи:

- відкликання користувацької ролі в SUA контролюється за допомогою співвідношення can_revoke_SUA ⊆ SAR × 2^R;

- група користувачів без відображення в UM керується за допомогою співвідношення can_revoke_UM ⊆ SAR × 2^G;

- відмова групової ролі в GA регулюється за допомогою співвідношення can_revoke_GA ⊆ SAR × 2^R.

Визначення 8. У моделі адміністративного відкликання групового рівня:

- відкликання користувацької ролі в GUA контролюється відповідно до відношення can_revoke_GUA ⊆ GAR × 2^R.

Зокрема, відношення can_revoke (x, {z}) вказує, що адміністративний елемент ролі x може відкликати користувача або групу з ролі чи групи в {z}. Як і в URA97, модель відкликання користувацької ролі (як на рівні системи, так і на рівні групи) GB-RBAC має два типи операцій: слабкий анулювання та сильний відкликання.

Зокрема, для (u, r) ∈ UA, слабка анотація на неї не має ефекту, якщо ви не прямо призначені для r, тобто існує r' ≤ r такий, що (u, r') ∈ UA; в той час як сильне відкликання на (u, r) намагається зробити слабке відкликання на (u, r') для кожного r' ≥ r, де (u, r') - явне призначення.

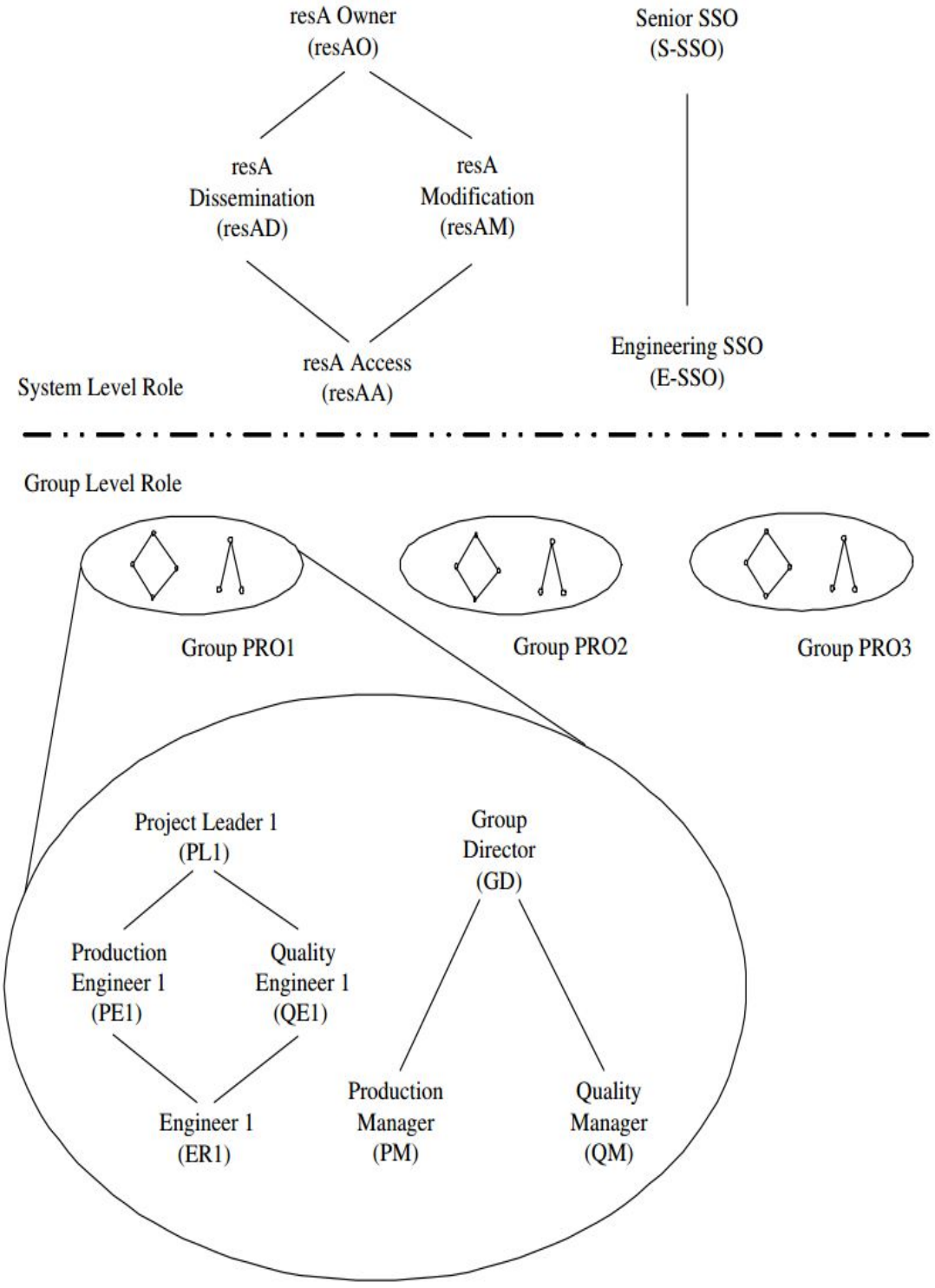

Рисунок 2.5 – Приклад: різні рівні ролей в GB-RBAC.

Таблиця 2.2 – Правила відкликання контролю

Тип	Адміністративна роль	Діапозон груп/ролей
can_assign_SUA	E-SSO	[resAA, resAD]
can_assign_UM	E-SSO	{@PRO1}
can_assign_GUA	PM	(ER1, PL1)

Розглянемо набір правил відкликання у таблиці 2.2 та інтерпретуємо його в контексті рисунка 2.5. Нехай Аліса буде членом ESSO, а Боб є членом resAD. Аліса спробує відкликати членство Боба від функції resAA: при слабкому скасування, це не має ефекту, оскільки Боб все ще є членом resAD; а з сильним відкликанням Аліса може відкликати Боба від resAA відповідно до першого правила в таблиці 2.2. Тепер ми розглядаємо відкликання на рівні групи. Нехай Аліса буде членом E-SSO, а Боб є членом групи PRO1 та ролі PE1. З правилом can_revoke_UM (E-SSO, @ PRO1), Аліса має право відкликати членство Боба з групи PRO1. З слабкою відміною, це не має ефекту, оскільки Боб все ще є членом PE1. Але це відміна не дозволяє Боба бути призначений для інших ролей у групі, наприклад, PL1 за GD (див. Таблицю 1). Сильне відкликання від членства Боба в PRO1 анулює Бобу всі члени ролей у PRO1. У групі PRO1 правило can_revoke_GUA (PM, (ER1, PL1)) вказує, що Керол, який є членом прем'єр-міністра, може відкликати Боба з PE1.

2.5.4 Переваги GB-RBAC над ARBAC97

GB-RBAC та його адміністративна модель є основною роботою для нашої схеми безпечної співпраці, показаної в наступному розділі. Ми підсумовуємо основні переваги нашого підходу, порівнюючи його з ARBAC97 наступним чином:

- Спрощений набір функцій користувача для системних адміністраторів У представленій моделі адміністратор повинен лише призначити користувача групі та вказати діапазон ролей групи. Після цього адміністратори групи беруть на себе

відповідальність за завдання користувача в цьому діапазоні ролей. Це значно спрощує завдання управління шляхом делегування адміністративних дозволів від централізованих адміністраторів системного рівня до децентралізованих адміністраторів групового рівня, особливо для динамічних та спеціальних групових додатків. На відміну від ARBAC97, коли в систему вводяться новий проект або новий департамент, створюється сукупність ролей та визначаються відповідні правила для його управління. Але в даній адміністративній моделі системному адміністратору потрібно лише створити групове відношення до групи, а інше призначення може бути кероване адміністраторами на рівні групи локально.

- Гнучке адміністрування для динамічного розподілу на користувачів. Адміністрація на рівні групи може легко підтримувати динамічну участь користувачів у групах. Наприклад, розглянута конференція Voice over IP (VoIP) повинна бути проведена в рамках групи PRO1 на рисунку 2.5. На підставі дозволів, наведених у таблиці 2.3, учасники PL1, PE1, QE1 та ER1 можуть приєднатися до цієї конференції, і учасники PL1, PE1 та QE1 можуть виступати на конференції, і лише учасники PL1 можуть приймати цю конференцію. Під час конференції користувачі можуть приєднатися та від’єднатися. З чисто адміністративним рівнем системи, є вимоги до динамічного розподілу призначених для користувача функцій, але це дуже ефективно з адмініструванням групового рівня. Наприклад, згідно з адміністративним правилом групового рівня в таблиці 2.2, будь-який член PM може керувати призначенням користувацької ролі в цій групі.

- Надзвичайно чітко визначена роль користувача. При ввімкненні GUA модель підтримує призначене завдання користувача на рівні групи. Як правило, адміністратор групи має більше контекстної інформації про дозволи та чутливі операції в групі та навички користувачів, отже, призначення користувачам на цьому рівні забезпечує точніший менеджмент користувачів та менший ризик отримання дозволів.

Таблиця 2.3 – Приклад розподілу ролей

Роль	Дозвіл	Роль	Дозвіл
PL1	conf1_host	PL2	conf1_host
PE1	conf1_speak(P1) prog1_upload(P2)	PE2	conf1_speak(P1) prog1_upload(P2)
QE1	conf1_speak prog1_report	QE2	conf1_speak prog1_report
ER1	conf1_join	ER2	conf1_join

- Налаштовувані адміністрації на рівні групи. Адміністратор на системному рівні може змінити призначення ролі групи і, таким чином, змінювати ролі, які адміністратор групи може призначити користувачам. Наприклад, згідно з правилами в таблицях 2.1 і 2.2, Аліса може змінювати статус учасників PRO1 від [ER1, PL1] до [ER1, PL1], відкликаючи (PRO1, PL1) з GA. Це значно забезпечує гнучкі та керовані адміністративні дозволи на рівні групи.

2.6 Підтримка тимчасового співробітництва з GB-RBAC

У цьому розділі спочатку визначаються загальні вимоги до контролю доступу для спеціальних взаємодій, а потім представляється рішення з GB-RBAC та запропоновано алгоритми для створення ролей на рівні груп та їхніх допусків.

2.6.1 Спеціальна схема співпраці

Спочатку визначається дві важливі особливості тимчасового співробітництва, які визначають вимоги до контролю доступу. Визначаємо автономний контрольний домен. Схема співпраці повинна забезпечити автономію управління окремими групами та обмін інформацією між групами.

Пропонується концепцію віртуальної групи. Ролі, що беруть участь у співпраці, експортуються у віртуальну групу з їх оригінальних (вихідних) груп.

Таким чином, більшість порушень/проблем вирішуються, наприклад, порушення і обмеження, такі як індукована SoD, усуваються на етапі призначення користувацької ролі.

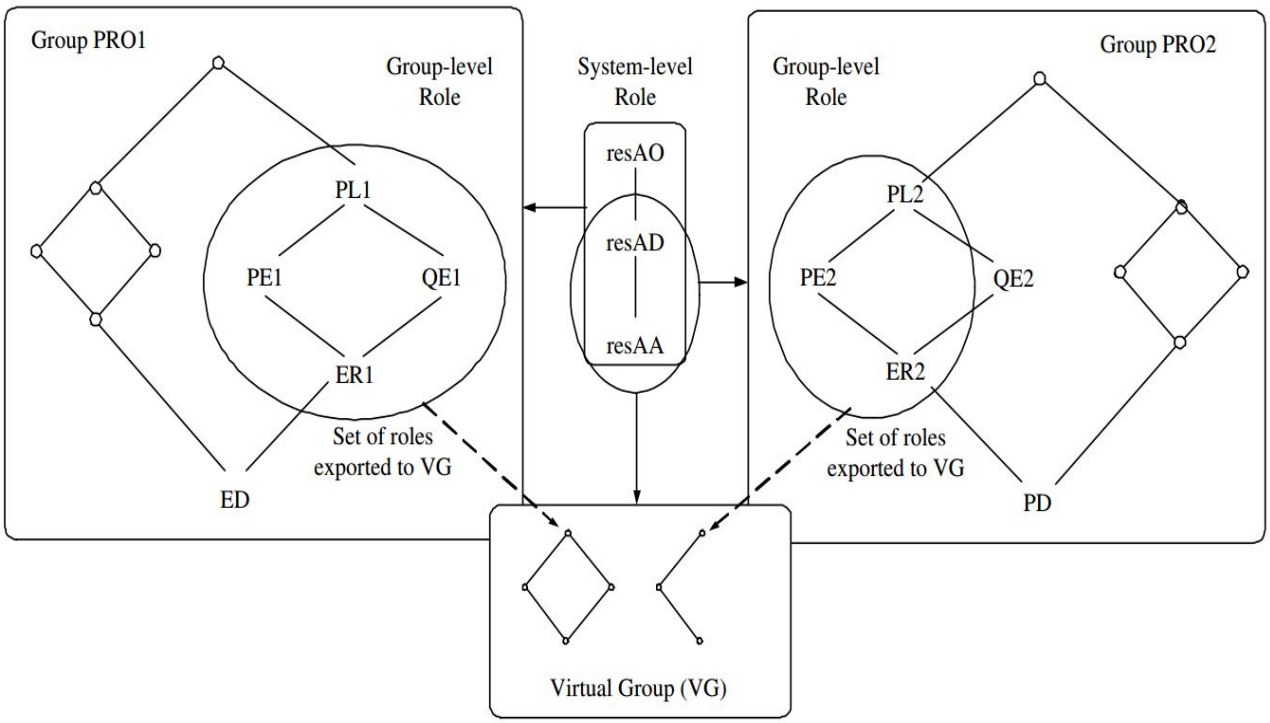

Рисунок 2.6 – Співпраця між групами з GB-RBAC.

Для підтримки спеціальної співпраці з GB-RBAC пропонується спеціальна група: віртуальна група (VG). VG має подібні функції, як загальні групи, за винятком того, що вона містить лише компоненти групового рівня (ролі та дозволи), експортовані з груп, що співпрацюють. На рисунку 2.6 ілюструється застосування VG.

Для співпраці між кількома спільними групами процедура будівництва стисло реалізується наступними кроками:

1. Запит на співпрацю надсилається спільним групам адміністратором вихідної групи (називається засновником VG), а відповідь надсилається назад до засновник.

2. За запитом адміністратори всіх спільних груп створюють VG за допомогою алгоритму ColGrant. У алгоритмі ColGrant ролі та дозволи спільних груп експортуються в VG на основі політики співпраці.

3. Адміністратори VG обираються від адміністраторів спільної групи. Для простоти, у цій статті ми припускаємо, що всі адміністратори групової роботи є адміністраторами VG.

4. Адміністратор VG може виконувати розподіл на роль користувача адміністративною моделлю користувача.

5. Всі члени VG можуть розпочати спільну роботу, а деякі модифікації співпраці можуть бути реалізовані за допомогою алгоритму ColUpdate. У алгоритмі ColUpdate ролі та дозволи в VG оновлюються відповідно до вимог змінених політик співпраці.

6. Закінчується спільна робота, а адміністратори спільних груп, які виходять з VG, остаточно знищують VG з алгоритмом ColRevo. У алгоритмі ColRevo ролі та дозволи відкликані з VG.

На рисунку 2.6 PRO1 експортує {ER1, PE1, QE1, PL1} у VG, а PRO2 експортує {ER2, PE2, PL2} у VG. Таким чином, VG містить ролі {ER1, ER2, PE1, QE1, PE2, PL1, PL2} та відповідні дозволи. Таким чином, всі члени, призначені для ролей у VG, можуть виконувати роль відповідних ролей у спільній роботі, незалежно від того, звідки їх оригінал.

Перш ніж визначити три алгоритми для основних операцій у співпраці, визначається конфлікт ролей, який може існувати при експорті ролей в різні групи до VG, а також визначення механізму іменування ролей, який використовується для вирішення конфліктів.

2.6.2 Рольові конфлікти

У схемі співпраці розглядається два типи конфліктів: конфлікт назв і конфлікт ролей. Інтерпретація назви відбувається при експорті ролі в VG, що має таку ж

назву іншої ролі у VG. Конфлікт ролі – це сучасна концепція та має наступне визначення:

Визначення 9. Роль r_j конфліктів з i_n у віртуальній групі VG, якщо $\exists p_1, p_2, p_3, p_4 \in P$, $(p_1, p_2) \subseteq$ дозволи$(r_i) \wedge (p_3, p_4) \subseteq$ дозволи$(r_j) \wedge$ пов'язані$(r_i, VG_y) \wedge \neg$ пов'язаний з $(r_j, VG) \wedge (p_1, p_3) \wedge \neg (p_2, p_4)$, де пов'язані з (r, VG) є предикатом, який перевіряє, чи r було експортовано в VG, і $\diamond$ означає, що два дозволи можуть бути отримані користувачем за допомогою ролі r_i і r_j одночасно.

Конфлікт ролі між r_i (у VG) і r_j (не в VG) трапляється, якщо існує два дозволи від r_i і r_j, які користувач може отримати, тоді як є ще два дозволи, які користувач одночасно не може отримати, наприклад, відповідно до заздалегідь визначеними обмеженнями щодо SoD. Це означає, що r_j є суперечливою роллю для VG. Наприклад, як показано на рисунку 2.7, P1 і P2 дозволу, що містяться в ролі QE2, P3 та P4, містяться в ролі PE1 (подробиці дозволу можна знайти в таблиці 2.3). Хоча P1 (conf1_speak) і P3 (conf2_speak) можуть одночасно досягатись користувачем (тобто користувач може говорити як на конференції 1, так і на конференції 2), P2 і P4 повинні бути виключно досягнуті користувачем, оскільки користувач не повинен одночасно мати дозволи для виконання операції завантаження програми та роботи над звітуванням програми відповідно до політики організації. Виходячи з визначення 9, існує конфлікт ролі між цими ролями.

Якщо існує конфлікт ролі, тоді потрібно розділити роль r_j на дві частини, наприклад, шляхом створення двох ролей та присвоєння P2 і P4 з QE2 відповідно до цих двох ролей. Для спрощення схеми ролевого співробітництва механізм іменування визначається наступним чином.

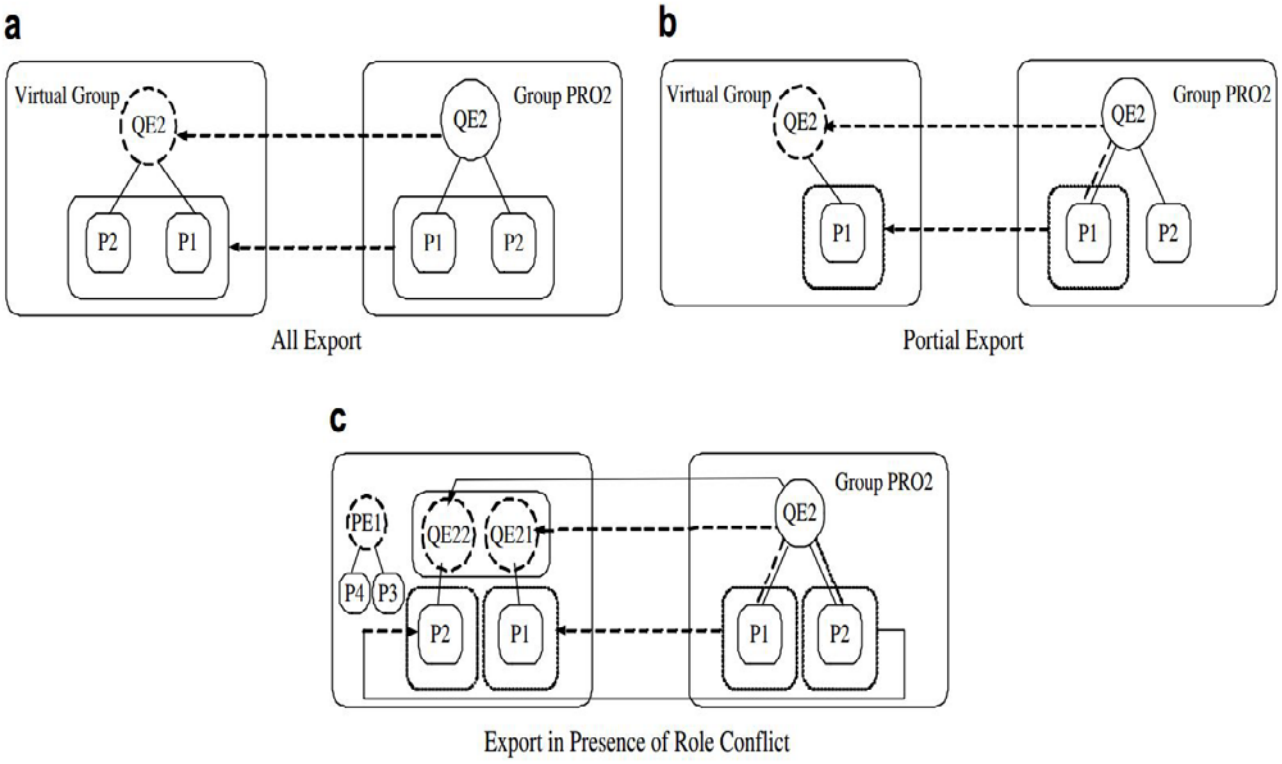

Рисунок 2.7 – Схема експорту ролей в GB-RBAC.

Визначення 10. Експортуючи компоненти спільної групи в віртуальну групу:

- якщо існує конфлікт імені, тоді нове ім'я ролі експорту – це його початкове ім'я плюс назву спільної групи;

- якщо існує конфлікт ролі, тоді нове ім'я конфліктуючої ролі – це його початкове ім'я плюс серійний номер кожної частини після розділення ролі.

Наприклад, якщо конфлікт імені існує, коли QE1 з PRO1 експортується в VG, ми називаємо роль QE1PRO1. Якщо існує конфлікт ролі, тоді розділяється конфліктуюча роль на дві частини. Наприклад, на рисунку 2.7, QE2 конфліктує з PE1, який вже експортується в VG. Тепер ми розділяємо QE2, і назва кожної частини QE2 називається QE21 і QE22. Загалом, ми розглядаємо три випадки, коли ролі та дозволи експортуються:

1. Ролі, які можна безпосередньо експортувати до VG.

2. Ролі, які можуть частково експортуватися в VG; тобто підмножину дозволів, отриманих ролями, можна експортувати.

3. Ролі, які повинні бути повністю експортовані в VG. Однак деякі з них суперечать існуючим ролям у VG, і набір дозволів, отриманих цими ролями, повинен експортуватися окремо.

На рисунку 2.7 показано сценарій, в якому QE2 PRO2 експортується до VG у різних випадках. У першому випадку експортуються всі дозволи QE2 на VG, і також можемо безпосередньо експортувати QE2. У другому випадку, лише підмножина дозволів QE2 може бути експортована до VG. Зокрема ми розділяємо дозволи QE2 та експортуємо частину з дозволу P1 на VG. У третьому випадку конфлікт QE2 і PE1 в VG. Розбиваються дозволи QE2 на підмножини P1 і P2, створюються дві нові ролі QE21 і QE22 і призначаються відповідно P1 і P2. Після цього, QE21 та QE22 експортуються відповідно до VG, а P1 і P2 також можуть бути експортовані до VG.

Усі користувачі та дозволи у віртуальній групі не містять жодної справжньої інформації. Вони є лише інформацією про зв'язок, яка позначає, який компонент надходить з якої групи. Проте роль у віртуальній групі присвоюється дозволами пов'язаної з нею ролі. Імена компонентів у віртуальній групі безпосередньо використовуються або походять від вихідних компонентів спільних груп.

2.6.3 Операції для співпраці

За допомогою визначених механізмів вирішення ролевого співробітництва представляємо три алгоритми, які використовуються для побудови віртуальної групи, оновлення компонентів віртуальної групи та знищення віртуальної групи, відповідно. Після створення віртуальної групи призначення користувачам ролей у групі може виконуватись адміністраторами групи за адміністративною моделлю.

Алгоритм ColGrant, показаний на рисунку 2.8, описує основні кроки побудови віртуальної групи серед спільних груп. У процесі побудови співпраці один з адміністраторів групи VG засновників створює назву віртуальної групи з необхідними параметрами, включаючи імена інших спільних груп. Алгоритми починаються шляхом експорту ролей і дозволів групи засновників у VG. Тут

розглянуто два випадки: якщо всі дозволи, включені в ролі груп, повинні бути експортовані, адміністратор безпосередньо експортує ролі та відповідні дозволи за допомогою функції ролі-посилання (r, VG_y); Якщо потрібно експортувати лише підмножину дозволів, алгоритм спочатку створює посилання ролі в контексті віртуальної групи, використовуючи функцію створення ролі, а потім вибирає дозволи, включені в ролі групи. Через функцію вставки (r, p) посилання на відповідні дозволи додаються в r, якщо тест буде успішним у експортованих дозволах за допомогою функції експорту-дозволу.

Якщо всі дозволи вставляються в нове посилання для ролі, посилання буде додано до VG за допомогою функції ролі-посилання. Після цього алгоритм досягає оновленого DSet і призначає користувачів віртуальній групі. Цей процес забезпечує створення віртуальної групи, а компоненти вихідної групи експортуються до віртуальної групи. Після подібного процесу адміністратори іншої групи учасників можуть експортувати необхідні ролі та дозволи віртуальній групі.

У кожному кроці ми перевіряємо, чи є ролі в оригінальній групі однаковими іменами з тими, які входять у віртуальну групу. Якщо є конфлікт назви ролі, ми змінюємо ім'я ролі за допомогою функції зміни імені та використовуємо модифіковане ім'я як ім'я ролі. Тепер розглянемо три випадки вищезгаданих і експортуємо ролі звуковим чином. Ми вже згадували перші два випадки на першому кроці. У третьому випадку ми створюємо дві нові ролі-посилання, використовуючи функцію створення ролі, і вставляємо посилання відповідних дозволів у відповідні ролі-посилання. Після додавання двох посилань на роль до VG ми оцінюємо DSet віртуальної групи, яка є об'єднанням DSet всіх спільних груп.

Тепер розглянемо кілька прикладів алгоритму ColGrant з рисунка 2.7. У першому стані ми припускаємо, що адміністратор PRO1 створює віртуальну групу (VG) і експортує всі ролі ER1, PE1, QE1 і PL1 та відповідні дозволи на VG. Оскільки досягаємо дозволів ролей через дозволи (r), ці процедури експорту реалізуються за допомогою функції "role-link".

ColGrant Algorithm

```
1)   DSet_tmp ← G_x.DSet
2)   if VG_y = ∅
3)       VG_y ← creategroup()
4)       VG_y = createVG();
5)       for each role r_i ∈ G_x.R_set
6)           if all-export(permissions(r_i))
7)               role-link(r_i,VG_y)
8)           else if part-export(permissions(r_i))
9)               r_new ← createrole()
10)              for each p_i ∈ permissions (r)
11)                  if export-permissions(p_i)
12)                      insert(p_i,r_new)
13)                      link-role(r_new,VG_y)
14)              if r_i ∈G_x.DSet
15)                  DSet_tmp ←DSet_tmp∪r_new - r_i
16)      VG_y.DSet ←VG_y.DSet ∪ DSet_tmp
17)  else
18)      for each role r_i ∈G_x.R_set
19)          if name-conflict(r_i,VG_y)
20)              r_i ← name-change(r_i)
21)-30)  similar with step 6)-15), we do not repeat it again
31)          else if permission-conflict(r_i, VG_y)
32)              permissions_con ← conflict-permissions(r_i, VG_y)
33)              r_new ← createrole()
34)              if export-permissions(permissions_con)
35)                  insert(permissions_con,r_new)
36)                  insert(permissions(r_i)-permissions_con, r_res)
37)                  link-role(r_new,VG_y)
38)                  link-role(r_con, VG_y)
39)              if r_i ∈ G_x.DSet
40)                  DSet_tmp ← DSet_tmp∪ r_new∪ r_res - r_i
41)      VG_y.DSet ← VG_y.DSet ∪ DSet_tmp
```

Рисунок 2.8 – Алгоритм ColGrant.

Припускається, що DSet в PRO1 - {ER1}, і цей набір об’єднується в DSet VG. Таким чином, DSet VG - {ER1}. На другому етапі PRO2 починає приєднуватися до VG за допомогою алгоритму. Оскільки конфлікту назви та ролі у поточному VG немає, то ролі {ER2, PE2, PL2} та відповідні права доступу безпосередньо

експортуються до VG. Вважається, що DSet з PRO2 є {ER2, PE2}, і цей набір об’єднується в DSet з VG, а значення DSet – {ER1, ER2, PE2}. За допомогою цих кроків завершується простий процес створення віртуальної групи. Адміністратори PRO1 і PRO2 стають адміністраторами віртуальної групи, і ці адміністратори можуть призначати ролі користувачам через can_ assign_GUA на адміністративній моделі групового рівня.

Таким чином, користувачі у віртуальній групі можуть мати дозвіл і почати спільну роботу один з одним.

Алгоритми ColUpdate (рисунок 2.9) і ColRevo (рисунок2.10) використовуються для оновлення компонентів віртуальної групи та видалення віртуальної групи відповідно. Оскільки процес додавання/видалення компонента в/з віртуальної групи аналогічний процесу в ColGrant, ці проблеми не представляються в алгоритмі ColUpdate. У алгоритмі ColUpdate використовується відмітка, щоб відрізняти різні випадки експорту ролей. Якщо дозволи ролі оновлюються, наприклад, у вихідній групі, тоді роль від'єднується та повторно експортується оновлена роль у віртуальну групу. У алгоритмі ColRevo також потребується відмітка, щоб розрізнити три різні випадки. Якщо роль безпосередньо експортується у віртуальну групу, посилання ролі видаляється. Однак, якщо роль експортується до віртуальної групи, вона розділена на дві ролі, тоді потрібно перетворити ім'я ролі та видалити посилання ролі. Якщо у віртуальній групі немає компонента, його можна знищити. Знову ж таки, ігнорується тут відмітка користувачів від ролей у віртуальній групі.

Посилання ролі в VG слід видалити, і роль повинна бути повторно експортована, коли ситуація експорту ролі відрізняється на різних етапах процесу ColUpdate. Наприклад, ми повинні розглянути справу про те, що дозволи на роль повністю експортуються в VG, а конфлікти ролей відбуваються в ColUpdate.

```
1)  if action = add
2)      similar with ColGrant Algorithm
3)  else if action = del
4)      similar with ColRevo Algorithm
5)  else if action = mod
6)      Initial Flag ← 0;
7)      if G_x.R_set ≠ ∅
8)      for each role r_i ∈G_x.R_set
9)          r_t ← name-change(r_i)
13)         if FindRole(r_t) = false
14)             r_t ← r_i
15)         if FindRole(r_t) = false
16)             Flag ← 1
17)         else
18)             r_new, r_res ← name-tranform(r_t)
19)             Flag ← 2
20)         if permission-conflict(r_t, VG_y)
21)             permissions_con ← conflict-permissions(r_t, VG_y)
22)             if Flag = 2
23)                 permission-update(r_new,permissions_con)
24)                 permission-update(r_res,
                      permissions(r_i)-permissions_con)
25)             else
26)                 role-unlink(r_t,VG_y)
27)                 r_new ← createrole()
28)                 insert(permissions_con,r_new)
29)                 insert(permissions(r_i)- permissions_con, r_res)
30)                 if (export-permissions(permissions_con)
31)                   link-role(r_new,VG_y)
32)                   link-role(r_con, VG_y)
33)         else
34)             if Flag = 2
35)                 role-unlink(r_new,VG_y)
36)                 role-unlink(r_res,VG_y)
37)                 role-link(r_t,VG_y)
38)             else
39)                 permission-update(r_t,permissions(r))
40)         if r ∈G_x.DSet
41)             update the role name in G_x.DSet
42)     update VG_u.DSet using G_x.DSet
```

Рисунок 2.9 – Алгоритм ColUpdate.

ColRevo Algorithm

1) $G_x.R_{set} \neq \emptyset$
2) **for** each role $r_i \in R_{set}$
3) $r_{tmp} \leftarrow$ name-change(r)
7) **if** FindRole(r_i) = false
8) $r_{tmp} \leftarrow r_i$
9) **if** FindRole(r_t) = false
10) Flag $\leftarrow 1$
11) **else if**
12) r_{new}, $r_{res} \leftarrow$ name-trnaform(r)
13) Flag $\leftarrow 2$
14) **if** Flag = 2
15) role-unlink(r_{new},VG_y)
16) role-unlink(r_{res},VG_y)
17) **else**
18) role-unlink(r_{tmp},VG_y)
19) **if** users(VG_y) = $\emptyset \wedge$ roles=(VG_y) = $\emptyset$
20) deleteVG()

Рисунок 2.10 – Алгоритм ColRevo.

2.7 Прототип реалізації та оцінки

Щоб показати доцільність та ефективність нашого підходу, впроваджується система-прототип шляхом покращення мови розмітки для розширеної контролю доступу (XACML) за допомогою моделі GB-RBAC.

2.7.1 Специфікація політики

Цей прототип використовує розширювану мову розмітки для контролю доступу (XACML) для визначення правил GB-RBAC. XACML – це формат відкритого стандарту для визначення правил контролю доступу та буде широко використовуватися з властивостями інтерпретації та розширюваності. Використовуючи Sun’s XACML бібліотеку, модуль керування політикою (PDP) інтерпретує політику XACML та приймає рішення.

У цьому прототипі політика дозволу та ролі користувача розміщується на платформі обслуговування авторизації. На рисунку 2.11 показано каркас двох зразкових полісів. У цих двох стратегіях PRO1_host є ім'ям ролі в PRO1. Перша політика стверджує, що ця роль має дозвіл на розміщення в групі PRO1 тільки тоді, коли жоден користувач не виступає в ролі хоста PRO1, або користувач має таку саму групу (PRO1) ідентифікатор, що і користувач, авторизований як PRO1_host одночасно який вказано доменом. Друга політика стверджує, що користувач, ім'я якого є Аліса в групі RPO1, призначається ролі PRO1_host лише в години з 9:00 до 19:00, що визначається PRO1.

Наслідок цих двох правил визначає користувача, ім'я якого є Аліса в рамках PRO1, має право запрошувати динамік в PRO1 на домен 1 тільки протягом 9:00 до 19:00, якщо жоден користувач не виступає в якості хоста PRO1, або користувач має ту ж групу (PRO1) ідентифікатор як і користувачів, авторизованих як PRO1_host.

2.6.2 Оцінка ефективності

Оскільки рішення GB-RBAC визначається шляхом перевірки облікових даних, запитуваних користувачем, об'єктів (ресурсів) та дій, слід враховувати ефективність процесів політики GB-RBAC. По-перше, оцінюється накладні витрати, введені децентралізованим управлінням у порівнянні з оригінальним XACML з профілем RBAC. По-друге, оцінюється продуктивність, коли авторизація користувача виконується на основі динамічно створюваних правил VG.

У проведеному експерименті запити на отримання дозволів створюються на платформах додатків та надсилаються авторизаційному серверу, який знаходиться в Java 1.4.2 і працює на комп'ютері під керуванням Windows XP з пам'яттю 1,7 ГГц Pentium M та 512 МБ. На рисунку 2.12 показано продуктивність запиту на авторизацію з оригінальним XACML, вказаним у профілі RBAC, і покращено XACML з профілем GB-RBAC. Час для процесу RBAC політики варіюється від 0 до 16мкс, а час обробки за авторизацією з розширеним XACML, визначеним

профілем GBRBAC, коливається від 0 до 47 мкс. Середній час виконання політики для дозволу за допомогою політики RBAC складає близько 7.92 мкс, а середній час роботи з GB-RBAC становить 9.95 мкс, що означає, що децентралізований механізм контролю доступу вводить приблизно 25% додаткових витрат. Оскільки авторизація користувача є одноразовою операцією, коли запит на авторизацію генерується з серверів додатків, ці витрати на додану вартість, введені децентралізованою адміністрацією RBAC, є розумними.

```
<PolicySet PolicySetId="Domain1:Role:Permission">
  ...
  <Subjects>...urn:mynamespace:role:PRO1\_host...</subjects>
  <Resources> ... invite_speaker ... </Resources>
  <Actions> ... PRO1 ... </Actions>
  ...
  <Condition>
    <EnvironmentAttributeDesigner AttributeID="group-id"/>
    <AttributeVaule>equal</AttributeValue>
  </Condition>
  ...
</PolicySet>


<PolicySet PolicySetId="PRO1:User:Role">
  ...
  <Subjects>
    <SubjectMatch>Alice</SubjectMatch>
    <SubjectMatch>urn:mynamespace:group:PRO1</SubjectMatch>
  </Subjects>
  <Resources>...urn:mynamespace:role:PRO1\_host...</Resources>
  <Actions> ... membership ... </Action>
  ...
    ...
  <Condition>
    ...
      <EnvironmentAttributeDesigner AttributeID="time"/>
      <AttributeVaule>9AM</AttributeValue>
    ...
      <EnvironmentAttributeDesigner AttributeID="time"/>
      <AttributeVaule>7PM</AttributeValue>
    ...
  </Condition>
  ...
<PolicySet>
```

Рисунок 2.11 – Політика щодо дозволів та рольових призначень у прототипі.

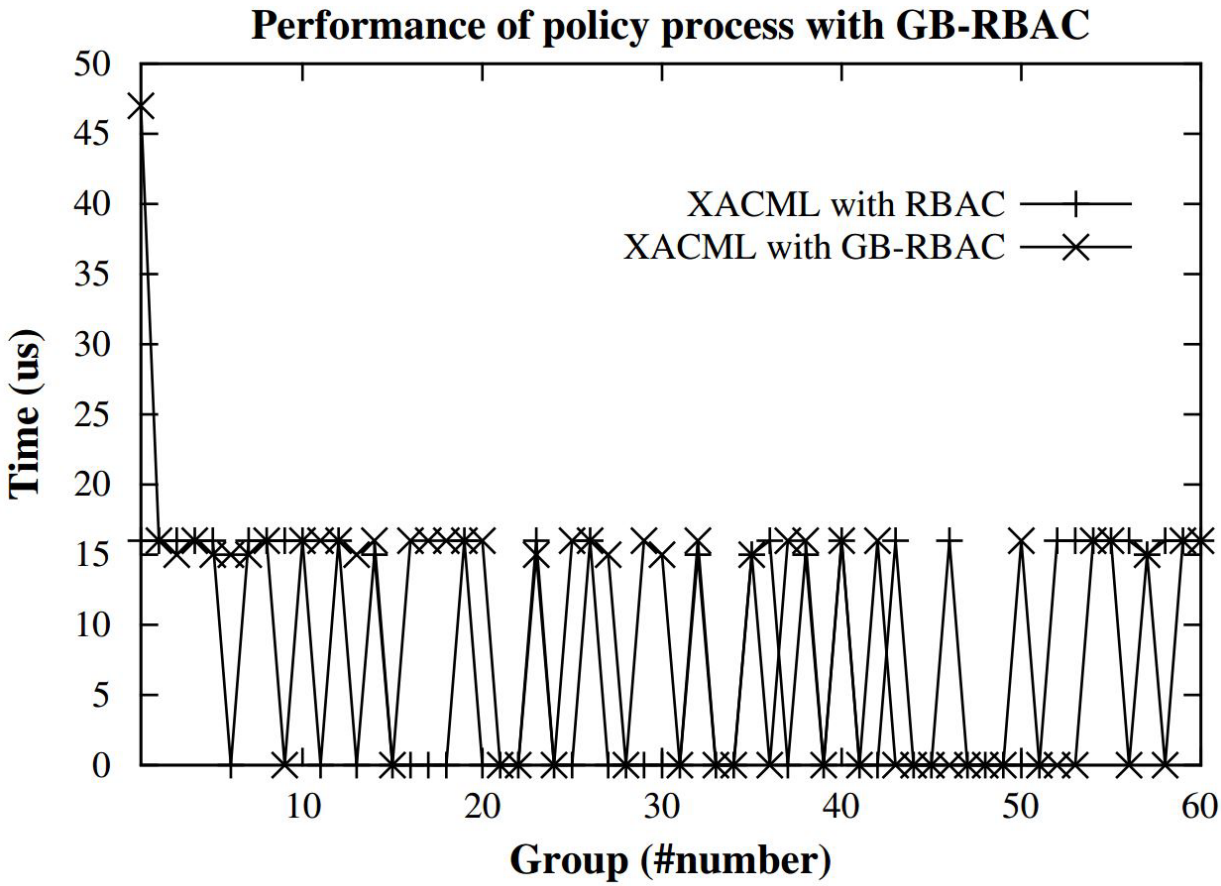

Рисунок 2.12 – Виконання процесу розробки політики з 60 учасниками групи

На рисунку 2.13 показані результати виконання політичного процесу без VG, з 1 VG та 60 VG, відповідно. Операція авторизації виконується наступним чином:

- процес політики без VG оцінює політику на основі статичної політики GB-RBAC, яка визначена на рисунку 2.11, і полягає в авторизації одночасних запитів для 60 учасників групи з зазначеної групи;

- процес політики з 1 VG оцінює політику, яка динамічно створюється для VG1, на основі політики, зазначеної на рисунку 2.11, і полягає у авторизації одночасних запитів для 60 членів групи з VG1;

- процес розробки політики з 60 VG оцінює політику, яка динамічно генерується для 60 VG, названих з VG1 на VG60, і є авторизацією одночасних запитів для 60 учасників з 60 різних віртуальних груп.

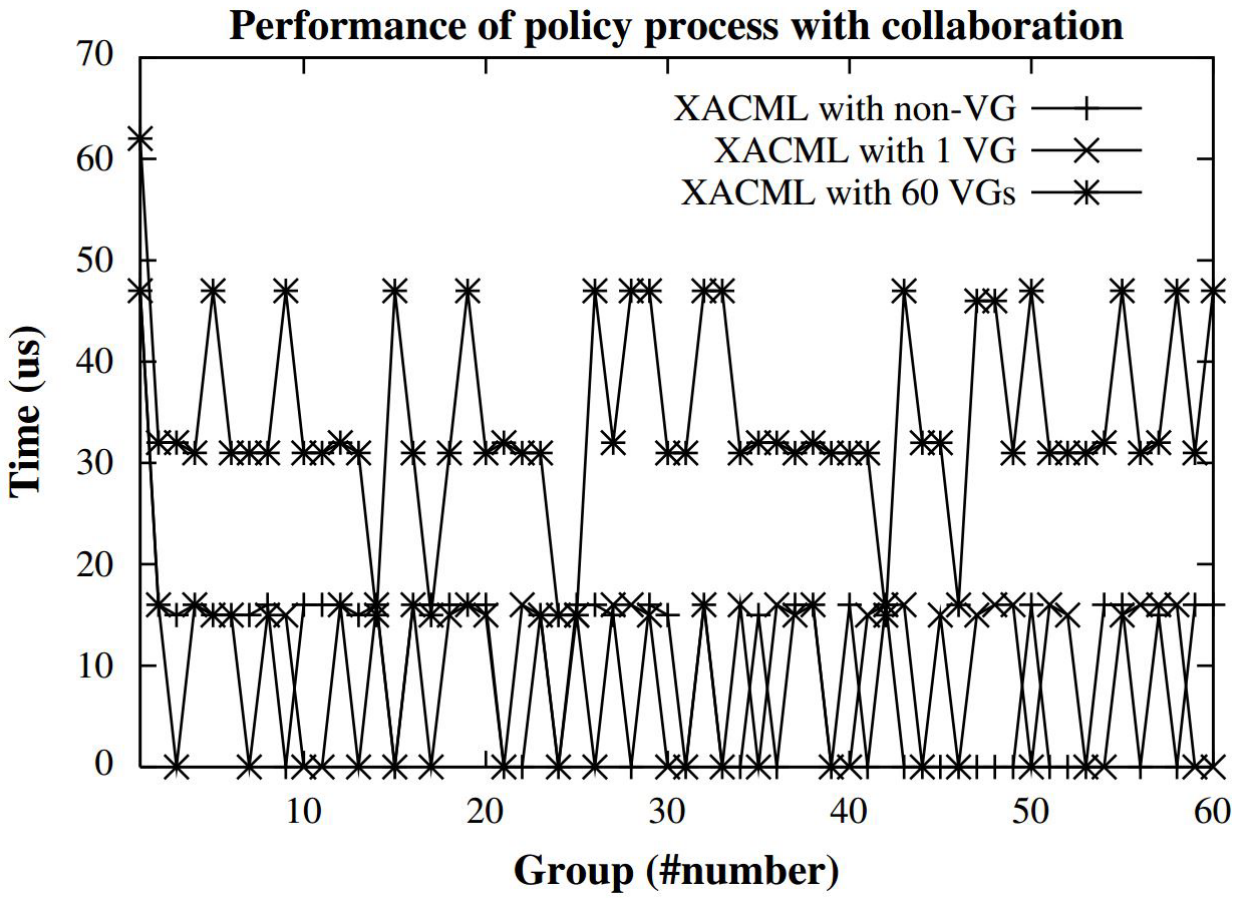

Рисунок 2.13 – Виконання політичного процесу з/без VG.

Для простоти, але не втрачаючи загальності, створюються тільки ті самі правила GB-RBAC для кожної групи членів у VG, а політики VG створюються лише за допомогою статичної групової політики, яка запускається адміністраторами різних груп джерел. Як показано на рисунку 2.13, час обробки генеруванням 1 VG становить близько 20 мкс, а середній час обробки для авторизації користувача становить 10.13 мкс, що лише забезпечує накладні витрати менш ніж на 1% порівняно з тим, що не мають генерації VG. Час авторизації користувача з генерацією 60 різних VG становить 34.36 мкс. Оскільки формування політики VG викликається лише першими користувачами VG (адміністраторами VG) і не буде вводити накладні витрати на авторизацію членів групи. Крім того, у більшості реальних застосувань, контроль доступу перевіряється лише один раз протягом певного типу безперервних операцій. Наприклад, читання партії файлів з каталогу

лише перевіряє, чи має користувач на початку права читання каталогу. Тому авторизація є прийнятною для забезпечення типових розподілених систем.

2.8 Висновки

Виходячи з адміністративної моделі Sandhu, розглянуто більш повне децентралізацію управління присвоєнням користувацької ролі, а наша модель вирішує проблему контролю доступу в програмах групового зв'язку, надаючи два способи призначень для користувача-ролі. Крім того, наша модель підтримує адміністрацію автономії, яка забезпечує легкий та універсальний спосіб для різних статичних та динамічних завдань.

Наша дворівнева адміністративна модель покращує існуючу модель у цьому аспекті та забезпечує гнучке та масштабоване керування у спільних обчислювальних середовищах. Зокрема, адміністратори системного рівня визначають функції ролі, а адміністратори на рівні групи призначають ролі користувача за локальними груповими політиками. Таким чином, наша модель підтримує дрібно-дзеркальні адміністрації відповідно до місцевої адміністративної політики, яка називається адміністративною автономією, яка є основою для адміністрації RBAC

У нашій схемі безпечного співробітництва ми пропонуємо підхід зверху донизу для об'єднання політики RBAC різних груп, тому наша схема дозволяє уникнути деяких проблем, які вносяться в підходи до відображення ролі, наприклад, ролеві порушення та порушення ролевих завдань. Компонент віртуальної групи введено, щоб уникнути прямого відображення ролей. Таким чином, більшість згаданих вище проблем усуваються.

3 РОЗДІЛ

ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ВПРОВАДЖЕННЯ СИСТЕМИ

Однією з головних цілей захисту інформаційних ресурсів від внутрішніх загроз є мінімізація збитків від порушення інформаційної безпеки підприємства.

Набирають оберти, такі структури, як логістичні центри. Вони займаються координуванням, аналізом ринку, спілкуванням з клієнтами, підтримкою клієнтів, оформленням замовлень та контролюванням процесу від покупки до доставки. Одна з таких компаній – «Jfront», яка є одною з лідерів на території України, але працює зі Сполученими Штатами Америки. А саме у сфері продажу автомобільних запчастин по території США.

Економічно доцільним слід вважати ситуацію, коли витрати на забезпечення інформаційної безпеки не перевищують збитків від реалізації загрози її порушення.

Щоб обґрунтувати економічну доцільність впровадження моделі GB-RBAC порівняємо величину витрат на впровадження GB-RBAC з величиною можливої шкоди, яку може понести підприємство внаслідок втрати інформаційних ресурсів.

3.1 Розрахунок (фіксованих) капітальних витрат

Капітальні інвестиції:

− вартість розробки проекту інформаційної безпеки (розробка схем пристроїв, політики функціонування системи тощо);

− вартість створення основного й додаткового програмного забезпечення (ПЗ);

− витрати на первісні закупівлі апаратного забезпечення;

− витрати на навчання технічних фахівців і обслуговуючого персоналу.

Спершу розрахуємо час, який буде витрачено на створення ПЗ:

$$t = t_{\text{тз}} + t_{\hat{a}} + t_{\acute{o}} + t_{\text{пр}} + t_{\text{опр}} + t_{\partial}, \text{ годин}, \qquad (3.1)$$

де $t_{\text{тз}}$ – тривалість складання технічного завдання на розробку ПЗ;

$t_{\hat{a}}$ – тривалість вивчення ТЗ, літературних джерел за темою тощо;

$t_б$ – тривалість розробки блок-схеми алгоритму;

$t_{пр}$ – тривалість програмування за готовою блок-схемою;

$t_{опр}$ – тривалість опрацювання програми на ПК;

$t_д$ – тривалість підготовки технічної документації на ПЗ.

Умовна кількість оперантів у програмі:

$$Q = q \cdot c\,(1 + p), \text{ штук,} \qquad (3.2)$$

де q – очікувана кількість оперантів - 120;

c – коефіцієнт складності програми -1.5;

p – коефіцієнт корекції програми в процесі її опрацювання – 0.05.

$$Q = 120 \cdot 1{,}5(1+0.05) = 189, \text{ штук.}$$

Оцінка тривалості складання технічного завдання на розробку ПЗ $\mathbf{t_{тз}}$ – 8 год.

Тривалість вивчення технічного завдання:

$$t_в = \frac{Q \cdot B}{(75...85) \cdot k} = \frac{189 \cdot 1.3}{80 \cdot 1.1} = 3, \text{ години,} \qquad (3.3)$$

де B – коефіцієнт збільшення тривалості етапу внаслідок недостатнього опису завдання, B = 1,3;

k – коефіцієнт, що враховує кваліфікацію програміста і визначається стажем роботи за фахом

- від 3 до 5 років – 1.1..1.2;

Тривалість розробки блок-схеми алгоритму:

$$t_б = \frac{Q}{(20...25) \cdot k} = \frac{189}{23 \cdot 1.1} = 7, \text{ годин.} \qquad (3.4)$$

Тривалість складання програми за готовою блок-схемою:

$$t_{пр} = \frac{Q}{(20...25) \cdot k} = \frac{189}{23 \cdot 1.1} = 7, \text{ годин.} \qquad (3.5)$$

Тривалість опрацювання програми на ПК:

$$t_{\text{опр}} = \frac{1{,}5Q}{(4...5)\cdot k} = \frac{1.5\cdot 189}{4\cdot 1.1} = 64 \text{, години.} \quad (3.6)$$

Тривалість підготовки технічної документації на ПЗ:

$$t_д = \frac{Q}{(15...20)\cdot k} + \frac{Q}{(15...20)}\cdot 0{,}75 = \frac{189}{17\cdot 1.1} + \frac{189}{17}\cdot 0.75 = 16 \text{, годин.} \quad (3.7)$$

$$t = 8 + 3 + 7 + 7 + 64 + 16 = 105, \text{ годин.}$$

Розрахунок витрат на створення програмного продукту

$$К_{пз} = З_{зп} + З_{мч}. \text{ грн} \quad (3.8)$$

Заробітна плата виконавця враховує основну і додаткову заробітну плату, а також відрахування на соціальне потреби (пенсійне страхування, страхування на випадок безробіття, соціальне страхування тощо) и визначається за формулою:

$$З_{зп} = t \cdot З_{пр} = 105 \cdot 297 = 31185 \text{, грн,} \quad (3.9)$$

де t – загальна тривалість створення ПЗ, годин;

$З_{пр}$ – средньогодинна заробітна плата програміста з нарахуваннями, грн/годину.

$$З_{пр} = \frac{З_м}{168} = \frac{50000}{168} = 297 \text{, грн/годину.} \quad (3.10)$$

де $З_м$ – середня заробітна плата на місяць – 50000 грн.

Вартість машинного часу для налагодження програми на ПК визначається за формулою:

$$З_{мч} = t_{опр} \cdot С_{мч} + t_д \cdot С_{мч} = 64\cdot 1.1 + 16\cdot 1.1 = 88 \text{, грн.} \quad (3.11)$$

де $t_{опр}$ – трудомісткість налагодження програми на ПК, годин;

$t_д$ – трудомісткість підготовки документації на ПК, годин;

$С_{мч}$ – вартість 1 години машинного часу ПК, грн./година.

Вартість 1 години машинного часу ПК визначається за формулою:

$$C_{\text{мч}} = P \cdot C_e + \frac{Ф_{зал} \cdot H_a}{F_p} + \frac{К_{лпз} \cdot H_{апз}}{F_p} = 0.4 \cdot 1.68 + \frac{10000 \cdot 0.1}{1920} = 1.192, \text{ грн/год,} \quad (3.12)$$

де Р – встановлена потужність ПК, 0.4 кВт;

$С_е$ – тариф на електричну енергію, 1.68 грн/кВт·година;

$Ф_{перв}$– первісна вартість ПК на початок року, 10000 грн.;

$Н_а$ – річна норма амортизації на ПК, 0.1 частки одиниці;

$Н_{апз}$ – річна норма амортизації на ліцензійне програмне забезпечення, частки одиниці;

$К_{лпз}$ – вартість ліцензійного програмного забезпечення, грн.;

$F_р$ – річний фонд робочого часу (за 40-годинного робочого тижня $F_р$ = 1920 год).

Отже:

$$К_{пз} = 31185 + 88 = 31272 \text{ грн.}$$

Таким чином, капітальні (фіксовані) витрати на проектування та впровадження проектного варіанта системи інформаційної безпеки складають:

$$К = К_{пз} + К_{аз} + К_{навч} + К_{н}, \text{ тис. грн} \quad (3.13)$$

де $К_{пз}$ – вартість створення програмного продукту, тис. грн;

$К_{аз}$ – вартість закупівлі апаратного забезпечення та допоміжних матеріалів, тис. грн;

$К_{навч}$ – витрати на навчання технічних фахівців і обслуговуючого персоналу, тис. грн;

$К_{н}$ – витрати на встановлення обладнання та налагодження системи інформаційної безпеки, тис. грн.

$К_{аз}$ = 10000 грн.

Витрати на навчання технічних фахівців і обслуговуючого персоналу, це є підготовчі курси з адміністрування та обслуговування системи виявлення вторгнень що складають 30 тис. грн;

$К_{навч}$ = 30000 грн.

Витрати на встановлення обладнання та налагодження системи інформаційної безпеки складають, 10 тис. грн.

$К_{н}$ = 10000 грн.

$$К = 31273 + 10000 + 30000 + 10000 = 81273 \text{ грн.}$$

3.2 Експлуатаційні витрати:

$$С_к = С_н + С_а + С_з + С_{ел} + С_{тос} \quad (3.14)$$

де витрати на навчання адміністративного персоналу й кінцевих користувачів($С_н$). визначаються за даними організації з проведення тренінгів персоналу, курсів підвищення кваліфікації – 10 тис. грн.

Річний фонд амортизаційних відрахувань ($С_а$) визначається у відсотках від суми капітальних інвестицій за видами основних фондів і нематеріальних активів (ПЗ) – 20% або 16000 грн.

Річний фонд заробітної плати інженерно-технічного персоналу, що обслуговує систему інформаційної безпеки ($С_з$), складає:

$$С_з = З_{осн} + З_{дод} = 50000 \cdot 12 + 5000 \cdot 12 = 660000 \text{ грн.} \quad (3.15)$$

де $З_{осн}$, $З_{дод}$ – основна середня заробітна плата на 01.12.2017, грн на рік.

Вартість електроенергії, що споживається апаратурою системою інформаційної безпеки протягом року ($С_е$), визначається за формулою:

$$С_{ел} = Р \cdot F_р \cdot Ц_е = 1 \cdot 8760 \cdot 1.68 = 14717 \text{ грн,} \quad (3.16)$$

де Р – встановлена потужність апаратури інформаційної безпеки, кВт;

$F_р$ – річний фонд робочого часу системи інформаційної безпеки (визначається виходячи з режиму роботи системи інформаційної безпеки);

$Ц_е$ – тариф на електроенергію, грн/кВт·годин.

Витрати на технічне й організаційне адміністрування та сервіс системи виявлення вторгнень визначаються у відсотках від вартості капітальних витрат 2%. А саме:

$$С_{тос} = К \cdot 0.02 = 1625 \text{ грн}$$

$$Ск = 10000 + 16000 + 660000 + 14717 + 1625 = 702000. \text{ грн.}$$

3.3 Оцінка можливого збитку від атаки (злому) на вузол або сегмент корпоративної мережі

Кінцевим результатом впровадження й проведення заходів щодо забезпечення інформаційної безпеки є величина відвернених втрат, що розраховується, виходячи з імовірності виникнення інциденту інформаційної безпеки й можливих економічних втрат від нього. По суті, ця величина відображає ту частину прибутку, що могла бути втрачена.

Загалом можливо виділити такі види збитку, що можуть вплинути на ефективність комп'ютерної системи інформаційної безпеки (КСІБ):

− порушення конфіденційності ресурсів КСІБ (тобто неможливість доступу до них неавторизованих суб'єктів або несанкціонованого використання каналів зв'язку);

− порушення доступності ресурсів КСІБ (тобто можливість доступу до них авторизованих суб'єктів (завжди, коли їм це потрібно);

− порушення цілісності ресурсів КСІБ (тобто їхня неушкодженість);

− порушення автентичності ресурсів КСІБ (тобто їхньої дійсності, непідробленості).

Вихідні дані:

$t_п$ = 6 годин – час простою вузла або сегмента корпоративної мережі внаслідок атаки, годин;

$t_в$ = 3 годин – час відновлення після атаки персоналом, що обслуговує корпоративну мережу, годин;

$t_{ви}$ = 1 годин – час повторного введення загубленої інформації співробітниками атакованого вузла або сегмента корпоративної мережі, годин;

$З_о$ = 30000 грн – місячна заробітна плата обслуговуючого персоналу (адміністраторів та ін.) з нарахуванням єдиного соціального внеску, грн на місяць;

$З_с$ =10000 грн – місячна заробітна плата співробітника атакованого вузла або сегмента корпоративної мережі з нарахуванням єдиного соціального внеску, грн на місяць;

$Ч_о$ = 6 – чисельність обслуговуючого персоналу (адміністраторів та ін.), осіб.;

$Ч_с$ = 54 – чисельність співробітників атакованого вузла або сегмента корпоративної мережі, осіб.;

О = 25 000 000 грн – обсяг чистого прибутку/дохід від реалізації/ атакованого вузла або сегмента корпоративної мережі, грн у рік, або оподаткований прибуток атакованого вузла або сегмента корпоративної мережі;

$П_{зч}$ = 8000 грн – вартість заміни встаткування або запасних частин, грн;

I=1 – число атакованих вузлів або сегментів корпоративної мережі;

N = 40 – середнє число можливих атак на рік.

Упущена вигода від простою атакованого вузла або сегмента корпоративної мережі становить:

$$U = П_п + П_в + V, \text{ грн.} \qquad (3.17)$$

де $П_п$ – оплачувані втрати робочого часу та простої співробітників атакованого вузла або сегмента корпоративної мережі, грн;

$П_в$ – вартість відновлення працездатності вузла або сегмента корпоративної мережі (переустановлення системи, зміна конфігурації та ін.), грн;

V – втрати від зниження обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі, грн.

Втрати від зниження продуктивності 54 співробітників з ЗП атакованого вузла або сегмента корпоративної мережі являють собою втрати їхньої заробітної плати (оплата непродуктивної праці) за 10 годин простою внаслідок атаки:

$$П_п = \frac{\sum З_с * Ч_с}{F} \cdot t_п \ , \qquad (3.16)$$

де F – місячний фонд робочого часу (при 40-а годинному робочому тижні становить 160-176 ч).

$$П_п = \frac{\sum 10000 \cdot 54}{160} \cdot 10 = 33750 \text{ грн.}$$

Витрати на відновлення працездатності вузла або сегмента корпоративної мережі включають кілька складових:

$$П_в = П_{ви} + П_{пв} + П_{зч}, \text{ грн.} \qquad (3.17)$$

де $П_{ви}$ – витрати на повторне уведення інформації, грн;

$П_{пв}$ – витрати на відновлення вузла або сегмента корпоративної мережі, грн;

$П_{зч}$ – вартість заміни устаткування або запасних частин, грн.

Витрати на повторне введення інформації $П_{ви}$ розраховуються виходячи з розміру заробітної плати 10000 грн 54 співробітників атакованого вузла або сегмента корпоративної мережі $З_с$, які зайняті повторним введенням втраченої інформації, з урахуванням необхідного для цього часу $t_{ви}$=1:

$$П_{ви} = \frac{\sum 10000 \cdot 54}{160} \cdot 1 = 3380 \text{ грн.} \qquad (3.18)$$

Витрати на відновлення вузла або сегмента корпоративної мережі $П_{пв}$ визначаються часом відновлення після атаки $t_в$ = 3 і розміром середньогодинної заробітної плати обслуговуючого персоналу (адміністраторів):

$$П_{пв} = \frac{\sum 30000 \cdot 6}{160} \cdot 3 = 3375 \text{ грн.} \quad (3.19)$$

$$П_в = 33750 + 3380 + 3375 = 40505 \text{ грн.}$$

Втрати від зниження очікуваного обсягу продаж в 25 000 000 грн за 10 годин простою атакованого вузла або сегмента корпоративної мережі виходячи із середньогодинного обсягу продажів і сумарного часу простою атакованого вузла або сегмента корпоративної мережі:

$$V = \frac{O}{F_г} \cdot (t_п + t_в + t_{ви}) = \frac{150000}{8760} \cdot (6 + 3 + 1) = 28500 \text{ грн ,}$$

(3.20)

де $F_г$ – річний фонд часу роботи організації становить близько 8760 ч.

$$U = П_п + П_в + V = 3380 + 40505 + 28500 = 72400 \text{ грн.}$$

Таким чином, загальний збиток від атаки на вузол або сегмент корпоративної мережі організації складе

$$B = \sum \sum U * N * I = 72400 \cdot 10 \cdot 1 = 724000 \text{ грн.} \quad (3.21)$$

3.4 Загальний ефект від впровадження системи інформаційної безпеки

Загальний ефект від впровадження системи інформаційної безпеки визначається з урахуванням ризиків порушення інформаційної безпеки і становить:

$$E = B \cdot R – C = 724000 \cdot 10 – 702000 = 6\,536\,500 \text{ грн,} \quad (3.22)$$

де В – загальний збиток від атаки на вузол або сегмент корпоративної мережі, грн;

R – очікувана імовірність атаки на вузол або сегмент корпоративної мережі, частки одиниці;

С – щорічні витрати на експлуатацію системи інформаційної безпеки, тис. грн.

Визначення та аналіз показників економічної ефективності системи інформаційної безпеки

Коефіцієнт повернення інвестицій ROSI:

$$ROSI = \frac{E}{K} = \frac{6536500}{81237} = 80{,}5, \text{ частки одиниці,} \qquad (3.23)$$

де Е – загальний ефект від впровадження системи інформаційної безпеки, грн;

К – капітальні інвестиції за варіантами, що забезпечили цей ефект, грн.

Термін окупності:

$$T_o = \frac{K}{E} = \frac{1}{ROSI} = \frac{1}{8{,}5} = 0.125, \text{ років.} \qquad (3.24)$$

3.5 Висновок

Розробка і впровадження моделі GB-RBAC для логістичної компанії «JFront» є економічно доцільним, так як витрати на її створення значно менші за суму збитків, завдяки не дорогій системи та мінімальній вартості комплектуючих необхідних для відновлення системи та її інформаційних ресурсів у разі успішних атак порушників. При цьому ми маємо:

- Капітальні витрати склали : К = 81237 (грн.);
- Поточні витрати склали : С = 702000 (грн.);
- Величина можливого збитку: $B = 724000$ (грн.);
- Загальний ефект від впровадження системи: Е = 6 536 500 (грн.);
- Рентабельність інвестицій у безпеку складає: ROSI = 8.5 (частки одиниці);
- Термін окупності капітальних інвестицій T_o = 0,125 (роки).

ВИСНОВКИ

У даній роботі представлено передові RBAC-моделі під назвою GBRBAC для безпечного співробітництва та адміністрування користувачів. Головною перевагою моделі є зручність та гнучкість для адміністрування у великих середовищах. Модель не повинна використовуватись у високоцентрованому контрольованому середовищі, а також надає два рівні адміністративних моделей для призначення користувальницьких функцій та зменшує складність адміністрування систем RBAC.

Крім того, призначення користувальницьких функцій на адміністративній моделі групового рівня забезпечує гнучкий спосіб для задоволення потреб групової рівності, тобто користувачі з різних груп утворюють віртуальну групу для спілкування. Крім того, ми пропонуємо спеціальну схему співпраці в багатокористувацькому середовищі на основі моделі GB-RBAC.

Схема пропонує компонент віртуальної групи для забезпечення безпечного співробітництва між різними групами. Представляється три алгоритми для перетворення компонентів з груп взаємодії в віртуальні групи та дозволяємо їм отримувати доступ до загальних ресурсів та інформації.

Таким чином, схема забезпечує безпечне та просте рішення для підтримки одночасного співробітництва. Реалізовано прототип з SunXACML, і експериментальні результати демонструють ефективність та масштабованість авторизації.

СПИСОК ЛІТЕРАТУРИ

1. Al-Shaer Ehab and Hamed Hazem, Discovery of Policy Anomalies in Distributed Firewalls, in IEEE INFOCOMM'04, 2004.
2. Bandara Arosha K , Emil C Lupu, Jonathan Moffet, Alessandra Russo, A Goal-based Approach to Policy Refinement, in: IEEE Policy, 2004.
3. Barrere F., A. Benzekri, F. Grasset, R. Laborde, B. Nasser, SPIDERNet : A Security Policy Derivation tool for Networks, in 3rd IEEE Latina America Network Operations and Management Symposium, 2003.
4. Barrere F., A. Benzekri, F. Grasset, R. Laborde , B. Nasser, Inter-Domains policy negociation. in: IEEE Policy, 2003.
5. Bishop M., "Computer Security: Art and Science", ISBN 0-201-44099-7, ed. Addison-Wesley, 2003.
6. Burch J., E. Clarke, D. Long, K. McMillan, D. Dill, L. Hwang, Symbolic Model Checking: 1020 states and beyond, in Information and Computation, pp 142-170, 1992.
7. Cholvy L., F. Cuppens, Analysing consistency of security policy, IEEE Symposium on Security and Privacy, 1997.
8. Crook R., D. Ince, B. Nuseibeh, Modeling Access Policies using Roles in Requirements Engineering, Information and Software Technology, 2003. 136 R. Laborde et al. / Electronic Notes in Theoretical Computer Science 121 (2005) 117–142
9. Ferraiolo D. F., R. Sandhu, S. Gavrila, D.R. Kuhn, and R. Chandramouli, "A Proposed Standard for Role-Based Access Control", Proposed 4/4/2003 Draft NIST, 2003, URL: http://csrc.nist.gov/rbac.
10. Guttman J., Filtering postures : Local enforcement for global policies, IEEE Symposium on Security and Privacy, 1997.
11. Guttman J., A. Herzog, F. Thayer, Authentication and confidentiality via IPsec, 6th European Symposium in Computer Security ESORICS, 2000.
12. Hinrichs S., Policy Based Management : bridging the gap, in 15th Annual Computer Security Applications Conference (ACSAC 99), 1999.

13. Jackson D., Alloy: a lightweight object modeling notation, ACM Transactions on Software Engineering and Methodology (TOSEM), v.11 n.2, p.256-290, 2002
14. Jennings N.R., S. Bussmann, Agent based Control Systems, why are they suited to engineering complex systems?, IEEE Control Systems Magazine, vol 23, No 3, 2003.
15. Jensen K., An Introduction to the Theoretical Aspects of Coloured Petri Nets, In: J.W. de Bakker, W.-P. de Roever, G. Rozenberg (eds.): A Decade of Concurrency, Lecture Notes in Computer Science vol. 803, Springer-Verlag 1994, 230-272.
16. Knorr Konstantin, Multilevel Security and Information Flow in Petri Net Workflows, in: Proceedings of the 9th International Conference on Telecommunication Systems - Modeling and Analysis, Special Session on Security Aspects of Telecommunication Systems, 2001.
17. Kristensen L.M., S. Christensen, K. Jensen, The Practitioner's Guide to Coloured Petri Nets, International Journal on Software Tools for Technology Transfer, 2 (1998), Springer Verlag, 98-132.
18. L¨uck I., C. Sch¨afer, H. Krumm, Model-based Tool-Assistance for Packet-Filter Design In: IEEE Policy, LNCS 1995, pp. 120-136, Springer-Verlag, 2001.
19. Krzysztof Juszczyszyn, "Verifying Enterprise's Mandatory Access Control Policies with Coloured Petri Nets", in Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, June 2003.
20. L¨uck I., S. V¨ogel, H. Krumm, Model-based configuration of VPNs, in Proc. 8th IEEE/IFIP Network Operations and Management Symposium NOMS 2002, pages 589-602, 2002.
21. Meadows C., The NRL Protocol Analyzer: An Overview, in Journal of Logic Programming 26 (2), pp 113-131, February 1996.

22. Melliar-Smith P., J. Rushby, The Enhanced HDM system for specication and verication, in Proc. VerkShop III, Wat-sonville, CA, Feb. 1985, pp. 41-43, published as ACM Software Engineering Notes, Vol. 10, No. 4.
23. Moffett J. D., Control Principle and Role Hierarchies, 3rd ACM Workshop on Role Based Access Control, 1998.
24. Moffet J., M. Sloman, Policy Hierarchies for Distributed Systems Management, IEEE Journal on Selected Areas in Communications, 11, 9, 1993.
25. Moore B., E. Ellesson, J. Strassner, A. Westerinen, “Policy Core Information Model – Version 1 Specification”, RFC 3060, February 2001.
26. Nyanchama M., S. Osborn, The role graph model and conflict of interest, ACM Transactions on Information and System Security (TISSEC), vol. 2, 1999.
27. Owre S., J. Rushby, N. Shankar, PVS: A prototype verification system, Lecture Notes in Computer Science, Vol. 607 (1992), Springer-Verlag.
28. Peri R., “Specification and verification of security policies”, PhD Dissertation, University of Virginia, January 1996. R. Laborde et al. / Electronic Notes in Theoretical Computer Science 121 (2005) 117–142 137
29. Samarati P., S. De Capitani di Vimercati, Access Control: Policies, Models and Mechanisms, Foundations of Security Analysis and Design, LNCS 2171, Springer-Verlag. 2001.
30. Westerinen A., J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, S. Wald-busser, “Terminology for Policy-Based Management”, RFC 3198, November 2001.
31. Wies R., Using a Classification of Management Policies for Policy Specification and Policy Transformation, In Proc. of the 4th IFIP/IEEE Int. Symposium on Integrated Network Management,, 1995.
32. Wijesekera D., S. Jajodia, A propositional policy algebra for access control, ACM Transactions on Information and System Security (TISSEC), vol 6,2003.

33. Yavatkar R., D. Pendarakis, R. Guerin, “A Framework for Policy-based Admission Control”, RFC 2753, January 2000.
34. ANSI. American national standard for information technology role based access control, ANSI INCITS 359–2004, Feb. 2004.
35. Crampton J. Understanding and developing role-based administrative models. In: proceedings of 12th ACM conference on computer and communications security; 2005. p. 158–67.
36. Crampton J. Discretionary and mandatory access controls for role-based administration. In: proceedings of 20th annual IFIP WG 11.3 working conference on data and applications security; 2006. p. 194–208.
37. Crampton J, Loizou G. Administrative scope: a foundation for role-based administrative models. ACM Transactions on Information and Systems Security 2003;6(2):201–31.
38. Ferraiolo D, Sandhu R, Gavrila S, Kuhn D, Chandramouli R. Proposed NIST standard for role-based access control. ACM Transactions on Information and Systems Security 2001;4(3): 224–74.
39. Joshi J, Bhatti R, Bertino E, Ghafoor A. Access control language for multidomain environments. IEEE Internet Computing 2004: 40–50.
40. Kapadia A, AI-Muhtdai J, Campbell R, Mickunas D IRBAC 2000: secure interoperability using dynamic role translation. In: Technical Report: UIUCDCS-R-2000-2162; 2000.
41. Koch M, Mancini LV, Parisi-Presicce F. Administrative scope in the graph-based framework. In: proceeding of the 9th ACM symposium on access control models and technologies; 2004. p. 97–104.
42. Nissanke N, Khayat EJ. Risk based security analysis of permissions in rbac. In: proceedings of 2nd international workshop on information systems; 2004.
43. Nita-Rotaru C, Li N. A framework for role-based access control in group communication systems. In: proceedings of international workshop on security and parallel and distributed systems; 2004.

44. Nyanchama M, Osborn S. The role graph model and conflict of interest. ACM Transactions on Information and Systems Security 1999;2(1):3–33.
45. Oh S, Sandhu R, Zhang X. An effective role administration model using organization structure. ACM Transactions on Information and System Security 2006;9(2):113–37.
46. Osborn S, Guo Y. Modeling users in role-based access control. In: proceedings of 5th ACM workshop on role-based access control; 2000. p. 31–8.
47. Osborn S, Sandhu R, Munawer Q. Configuring role-based access control policies. ACM Transactions on Information and Systems Security 2000;3(2):85–106.
48. Park J, Sandhu R, Ahn GJ. Role-based access control on the web. ACM Transactions on Information and Systems Security 2001; 4(1):37–71.
49. Piromruen S, Joshi J. An RBAC framework for time constrained secure interoperation in multi-domain environments 2005:36–48.
50. Core and hierarchical role based access control (RBAC) profile of XACML v2.0, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf.
51. Sandhu R. Role versus group. In: proceeding of 1st ACM workshop on role-based access control; 1995. p. 1–12.
52. Sandhu R, Coyne E, Reinstein H, Youman C. Role-based access control model. IEEE Computer 1996;29(2):38–47.
53. Sandhu R, Bhamidipati V, Munawer Q. The ARBAC97 model for role-based administration of role. ACM Transactions on Information and Systems Security 1999;2(1):105–35.
54. Shafiq B, Joshi J, Bertino E, Ghafoor A. Secure interoperation in a multidomain environment employing RBAC poilcies. IEEE Transactions on Knowledge and Date Engineering 2005;17(11): 1557–77.

ДОДАТОК А. Перелік документів на оптичному носії

01 Титульна сторінка.docx;

02 Завдання.docx;

03 Реферат.docx;

04 Список умовних скорочень.docx;

05 Зміст.docx;

06 Вступ.docx;

07 Преший розділ.docx;

08 Другий розділ.docx;

09 Економ розділ.docx;

10 Висновки.docx;

11 Список використаної літератури.docx;

12 Додаток А.docx;

13 Додаток Б.docx;

14 Додаток В.docx;

15 Презентація.pptxx.

ДОДАТОК Б. Відгуки керівників розділів

Б.1 Відгук керівника економічного розділу

Керівник розділу _____ _____

(підпис) (ініціали, прізвище)